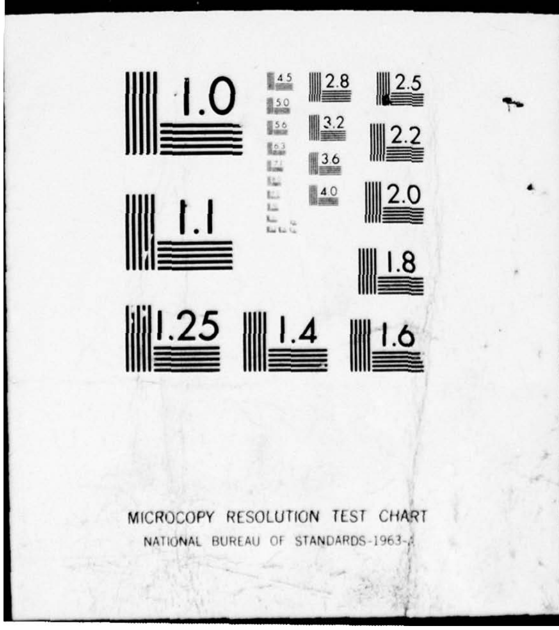
AD-A040 053 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM). VOLUME 2, SECTION--ETC(U)
DEC 71 G M CLARK, R J WILHELM DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-2(U)-SECT-1 NL

1 OF 5
AD
A040053

040053


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

REPORT RF 2995 FR 71-2 (U)
SECTION 1

EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM) PROGRAM DOCUMENTATION (Common Descriptions and Flow Charts of Subroutines ADJPOS through CXYLOC)

Final Report

Contract No. DAAH01-70-C-0713

~~This document is subject to Special Export Controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Headquarters, U.S. Army Missile Command, ATTN: DRSMI-CM~~

by

Gordon M. Clark
Robert J. Wilhelm

SYSTEMS RESEARCH GROUP
Department of Industrial and Systems Engineering
The Ohio State University
Columbus, Ohio 43210

for

SYSTEMS ANALYSIS OFFICE
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Volume #2, Section #1.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) R.-2995-FR-71-2 (U)-Section-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM), PROGRAM DOCUMENTATION. (Common Descriptions and Flow Charts of Subroutines ADJPOS through CXYLOC).		5. TYPE OF REPORT & PERIOD COVERED Final Report, (1/70 to 12/70)
		6. PERFORMING ORG. REPORT NUMBER Jan - Dec 70.
7. AUTHOR(s) (10) Gordon M. Clark, Robert J. Wilhelm		8. CONTRACT OR GRANT NUMBER(s) (15) DAAH01-70-C-0713
9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Research Group, Department of Industrial and Systems Engineering, The Ohio State University Columbus, Ohio 43210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11) 31 Dec 71
11. CONTROLLING OFFICE NAME AND ADDRESS Systems Analysis Office, DRSMI-CM U. S. Army Missile Command Redstone Arsenal, Alabama 35809		12. REPORT DATE December 31, 1971
		13. NUMBER OF PAGES 468
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 464p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

~~This document is subject to Special Export Controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Headquarters, U. S. Army Missile Command, ATTN: DRSMI-CM~~

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Armor	Combat Simulation	Subroutines	Firepower
Helicopters	Computer Simulation	Communications	Land Combat
Missiles	Programming	Detection	Tactical Doctrine
Crew-Served Weapons	Flow Charts	Probability	Environment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This volume contains program documentation for three significant extensions to the DYNCOM combat simulation. Subroutine flow charts and labeled common area descriptions are presented in two appendixes for models created to represent: 1) helicopter units in support of armored units up to battalion size; 2) semi-active guided missiles with a ballistic trajectory; and 3) movement of crew-served weapons. The volume is organized in two sections. This section, Section 1, contains all of the → next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

267 355

1B

19. Combat Effectiveness
Protection
Flight Dynamics
Intelligence
Target Selection
Firing
Lethality
Terrain
Fire Support
Command and Control

cont.
20. common area descriptions and flow charts of subroutines ADJPOS through CXYLOC. Section 2 contains the remainder of the flow charts.

FOREWORD

Research conducted by the Systems Research Group under Contract DAAH01-70-C-0713 with the U. S. Army Missile Command (MICOM), Systems Analysis Office, is reported in three volumes of which this volume is the second. The objective of this research is to extend DYNCOM to represent:

1. helicopter units,
2. ballistic-trajectory guided missiles,
3. movement of crew-served weapons,
4. electronic countermeasures, and
5. effects of smoke and haze on missile guidance systems.

Volume 1 presents the design of models to represent helicopters and ballistic-trajectory guided missiles. This volume, Volume 2, contains the DYNCOM program documentation (flow charts and common descriptions) for the helicopters, guided missile, and crew-served weapon models. The Classified Annex, Volume 3, documents the models to represent electronic countermeasures and the effects of smoke and haze.

The program documentation presented in this volume consists of two appendixes. Appendix A contains the description of labeled commons explicitly created and/or modified for the helicopter, ballistic trajectory guided missile, and crew-served weapon movement models. Appendix B contains flow charts of subroutines created and/or modified for these three models. The contents of both appendixes are arranged in alphabetical order with respect to the common or subroutine name. Because of the size of this volume, it appears in two sections. This section, Section 1, contains Appendix A and Appendix B up to subroutine CXYLOC. The remainder of Appendix B is in Section 2. In addition, Chapter 1 of this volume contains a summary of the changes to the Crew-Served Weapon Movement Model in order to implement it in DYNCOM.

Conclusions drawn in this report represent the current views of the Systems Research Group, The Ohio State University, and should not be considered as having official MICOM or Department of Army approval, either expressed or implied, until reviewed and evaluated by those agencies and subsequently endorsed.

The cooperation received from MICOM in preparing this report has been extremely helpful. In addition, we wish to acknowledge the advice provided, and the cooperation received, from the U. S. Army Combat Developments Command (USACDC). In particular, the Systems Analysis Group and the aviation agency have been most helpful.

We would like to acknowledge the important contributions of Lois Graber who patiently typed and proofread the text. We also extend our appreciation to the programmers, Robert J. Wilhelm, Charles McCartney, William Hess, and Gerald Petty, who assisted so ably in developing these extensions to the DYNCOM program. Without their contributions, this model would not have been developed.

TABLE OF CONTENTS

CHAPTER 1

CREW-SERVED WEAPON MOVEMENT MODEL

by
S. Parry

The primary mission of crew-served weapons is to provide the commander of a combat unit with organic assault and anti-tank capabilities. A crew-served weapon (CSW) unit is considered to be composed of one weapon, and the personnel and equipment required for operation of that weapon. The factors listed below highlight the basic differences between the armored vehicle and the CSW unit:

1. The dismounted CSW unit is more susceptible to enemy fire because of its limited movement speed and lack of protection;

2. CSW units are more difficult to detect because they are generally man-portable and hence have a greater ability to attain and maintain concealment than does an armored vehicle;

3. Dismounted CSW units may be placed in strategic locations unattainable by armored vehicles because of their ability to avoid detection and to move over terrain that may be untrafficable for armored vehicles.

Research summarized in this chapter represents the third phase of research conducted by the Systems Research Group in the area of crew-served weapons. The initial phase of research, reported in reference 1, resulted in models to describe the following processes:

1. The process by which crews dismount from personnel carriers, establish defensive positions at the dismount point, and remount when appropriate;

2. The detection of dismounted crew-served weapon units; and

3. The infliction of damage upon crew-served weapon targets.

In addition, the research describes modifications to existing simulation models such as the Communications, Movement Controller, Fire Controller, and Firing Models.

The second phase of research, reported in reference 2, extended the initial research to more effectively evaluate the combat performance of crew-served weapons in armored-unit engagements. The following models and methodology were developed:

1. Methodology was developed to determine preferred firing positions for dismounted CSW units operating in either attack or defense modes;

2. The capability was developed to represent either single or multiple CSW units dismounting from a personnel carrier, and subsequent operations of the CSW units.

3. Given a desired firing position for a dismounted CSW unit, a dynamic route selection model was developed to select the optimal route of advance from the dismount point to the desired firing position in accordance with specified tactical doctrine.

4. A movement model was developed to move dismounted CSW units along their selected routes to their respective firing positions.

During the phase of research described in this chapter, the following items were accomplished:

1. Methodology was developed to dynamically move the dismounted CSW units back to their APC and to remount the crews (Subroutine MOUNT);

2. All existing CSW model subroutines were programmed in DYNCOM, and interface logic required for linkage with existing DYNCOM routines was developed;

3. Time coordination logic enabling the APC and dismounted CSW units to operate either independently or together was developed;

4. The CSW model was successfully run with real data in DYNCOM.

The logic flow of the CSW routines is given in Figure 1 . When DYNCOM determines that the current element is either an APC which is to dismount or remount crews or a dismounted CSW unit, subroutine CSWCON, which controls the computations for CSW units, is called. If the element is an APC, primary and alternate desired firing positions, as well as the optimal route to the primary firing position, are determined as a function of the deployment mode for each CSW unit on the APC. If the element is a dismounted CSW unit, the movement path (either toward a firing position or the selected remount point) is determined

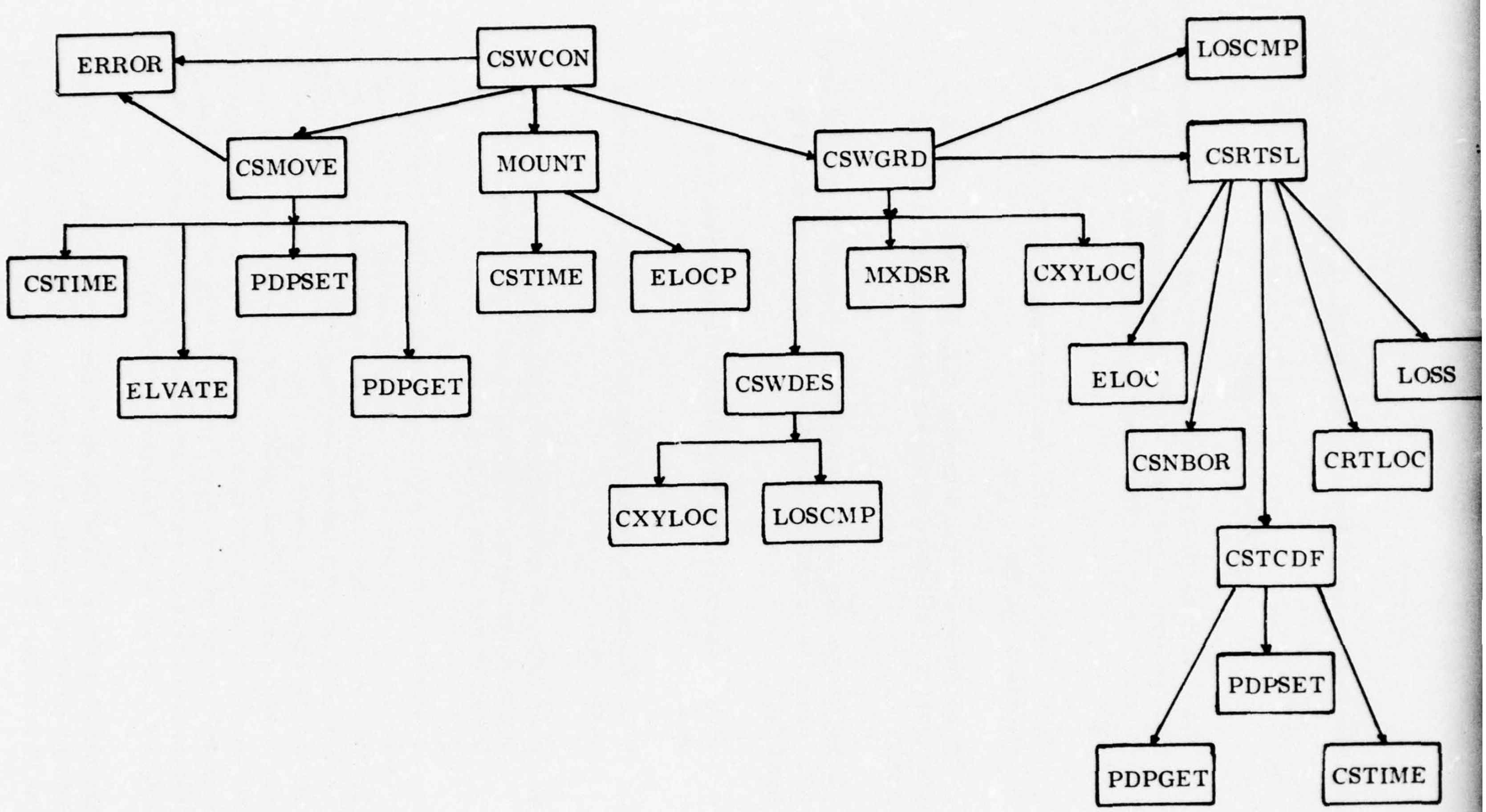


Figure 1.--CSW Routine Logic Flow

for the event. The dismounted CSW unit is then moved by using subroutine CSMOVE.

Deployment Modes

CSW units may be deployed in either the attack or defensive modes. The determination of the deployment mode for the CSW units depends on the location of the threat position for the APC. The threat position is determined according to the following logic:

1. If the APC has a target, the threat position (XT, YT) is set as the coordinates of the target.

2. If the APC does not have a target, (XT, YT) is determined as the centroid of all enemy elements currently detected by the APC.

3. The mode of deployment depends on the distance of the APC from XT, YT and is controlled by a distance, RMIN, specified by input data.

   a. If the distance is less than RMIN, the deployment mode is defensive;

   b. If the distance is greater than RMIN, deploy in the attack mode; and

   c. If no enemy elements are currently detected by the APC, deploy in the defensive mode facing in the direction of the maneuver unit's heading.

If the attack mode is utilized, a fire-position selection grid is constructed in the sector of responsibility. Each point in the grid is evaluated as to its desirability considering the portion of the target threat position covered relative to the grid point; the portion of the grid point covered or concealed relative to the target threat position; and the range-firepower index for the CSW unit relative to the grid point-target threat position combination. The primary desired firing position is selected as the grid point with maximum desirability within a specified allowable distance from the primary firing position. The purpose of selecting an alternate diring position is to provide the capability of representing "shoot and scoot" tactics on the part of the dismounted CSW unit.

An example of an actual DYNCOM run showing the selection of primary and alternate firing positions is given in Figure 2 . Note that the grid consists of 22 rows and 16 columns. The primary firing position (row 20, column 6)

```
**********ELEMENT NUMBER,    2 IS APC NUMBER    2 DEPLOYING IN THE ATTACK MODE   **********
 CSWGRD1     -0.10218    -0.10200    0.99478     1589.0     1200.0     100.0   22   16
              387.0       -39.7       1     8
```

*** THE DESIRABILITY GRID ***

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0				
0.0	0.3274	0.2879	0.2973	0.3010	0.1585	0.1119	0.3792	0.4691	0.2695
0.6087	0.3814	0.0545	0.5519	0.0	0.0				
0.2655	0.5614	0.4066	0.3694	0.3060	0.4282	0.3514	0.4323	0.2700	0.0728
0.0843	0.3820	0.3441	0.5166	0.2200	0.4533				
0.2731	0.4341	0.5067	0.6588	0.1116	0.2917	0.5293	0.5929	0.3987	0.3674
0.3481	0.4861	0.1310	0.2145	0.2318	0.4456				
0.2381	0.2533	0.1259	0.5795	0.5589	0.3207	0.2171	0.3022	0.4251	0.4021
0.5811	0.3191	0.4485	0.3973	0.3177	0.4450				
0.6302	0.2378	0.3629	0.4519	0.5709	0.3922	0.6167	0.1485	0.3654	0.3217
0.4392	0.4076	0.4257	0.4880	0.2360	0.2562				
0.5169	0.3937	0.5843	0.5136	0.3080	0.3004	0.2078	0.3079	0.4831	0.1889
0.4842	0.5356	0.4293	0.4874	0.3223	0.5915				
0.5327	0.3450	0.4039	0.5901	0.5083	0.4174	0.3573	0.2139	0.2995	0.3135
0.2086	0.4048	0.4137	0.3135	0.5001	0.4305				
0.2427	0.4928	0.3432	0.4242	0.5073	0.3725	0.3700	0.5372	0.5732	0.5933
0.3378	0.6243	0.2505	0.4420	0.6101	0.3203				
0.4490	0.4458	0.5110	0.6662	0.5479	0.7026	0.4577	0.2165	0.3301	0.5131
0.5278	0.4005	0.3889	0.6860	0.7406	0.7258				
0.5242	0.3904	0.6039	0.2339	0.4607	0.5421	0.4201	0.4177	0.4940	0.4655
0.6811	0.5076	0.5357	0.6561	0.5573	0.7089				
0.4801	0.4303	0.5456	0.6026	0.3235	0.5928	0.6341	0.5743	0.2560	0.4305
0.6180	0.6472	0.3150	0.6205	0.6736	0.5695				
0.5180	0.2567	0.2576	0.5252	0.4618	0.4842	0.7345	0.3671	0.7047	0.3056
0.4062	0.7682	0.2587	0.4147	0.5701	0.1947				
0.7318	0.4683	0.4487	0.3203	0.4554	0.4186	0.2677	0.3790	0.5307	0.5612
0.6873	0.7356	0.6159	0.7819	0.6752	0.3791				
0.3388	0.5824	0.5009	0.5537	0.4659	0.5239	0.3585	0.7916	0.3329	0.2735
0.7054	0.3315	0.6986	0.7306	0.6645	0.5148				
0.4803	0.7228	0.3619	0.3744	0.5733	0.5131	0.7242	0.7084	0.6226	0.7434
0.6127	0.5811	0.5252	0.5092	0.7879	0.2415				
0.7710	0.5627	0.5539	0.5752	0.7733	0.6755	0.6694	0.8004	0.5084	0.5485
0.4769	0.4435	0.8281	0.6130	0.5586	0.7625				
0.5918	0.3777	0.5257	0.4264	0.6486	0.5579	0.7360	0.6659	0.6665	0.4938
0.6131	0.7751	0.7362	0.4682	0.7258	0.5661				
0.6301	0.6841	0.6334	0.8457	0.6398	0.6148	0.3937	0.8109	0.7003	0.6417
0.5939	0.7000	0.2863	0.4043	0.6353	0.3647				
0.5605	0.5953	0.4910	0.6682	0.6457	$\boxed{0.8915}$ P	0.6845	0.6746	0.7963	0.6457
0.5704	0.5795	0.7524	0.6693	0.5402	0.4579				
0.3752	0.7926	0.6197	0.6287	$\boxed{0.8443}$	0.6854	0.5783	0.5923	0.7191	0.4014
0.6127	0.6945	0.6860	0.4387	0.5112	0.4028				
0.5440	0.5544	0.7989	0.7168	0.5342	0.5313	0.6166	0.4924	0.4814	0.7117
0.7017	0.5812	0.5371	0.4553	0.6064	0.6640				

```
POSITION  20,   6  HAS MAX DESIRE OF   0.8915
POSITION  21,   5  IS ALTERNATE WITH DESIRE OF   0.8443
```

Figure 2.--Attack Mode Deployment

has a desirability of 0.8443. In this example, equal weights of importance (0.33) were assigned to the cover factors and the range-firepower index.

If the defensive mode is utilized, deployment of the CSW units occurs along a line through the APC perpendicular to the threat position specified for the APC. The fire position selection grid for defensive deployment consists of points along the deployment line of a specified length. The desirability of each point is evaluated as in the attack mode, except that the range firepower index is not used.

An example of a defense mode deployment fire position grid is given in Figure 3. Note that the second column point is selected (with a desirability of 0.5913). In this case equal weights (0.5) were assigned to each cover factor.

The logic to determine firing positions for dismounted CSW units is accomplished by subroutines CSWGRD, CSWDES, MXDSR, CXYLOC, and LOSCMP (see Figure 1).

Route Selection

After the desired firing position has been selected for each dismounted CSW unit, a route from the dismount point to the firing position is selected. This logic is accomplished by subroutines CSRTSL, CSNBOR, CSTCDF, CRTLOC, ELOC, LOSS, PDPGET, PDPSET, and CSTIME (see Figure 1).

A square grid is established with orientation on the battlefield such that the dismount point is in the center of the second row and the firing position is in the center of the last row. The relative tactical difficulty of each grid point is determined by considering travel time, known enemy elements, and known enemy strong points. The optimal route is determined utilizing the dynamic programming algorithm which has been developed for the armored units.

An actual difficulty grid from which the optimal route is selected is given in Figure 4. The X-Y coordinates of the route actually selected from the difficulty grid are shown in Figure 5.

Movement of Dismounted CSW Units

The CSW Route Selection Model generates a sequence of movement control points through which dismounted CSW units are to move when deploying in the attack mode. The use of movement control points and plane departure points is adapted fromtheir use by armor units in DYNCOM (see Movement, Chapter 7 in reference 3).

The movement speed for a dismounted CSW unit is determined as a function of the grade angle of the terrain being traversed and the unit's weapon code. The

```
**********ELEMENT NUMBER,    1 IS APC NUMBER    1 DEPLOYING IN THE DEFENSE MODE **********

                              *** THE DESIRABILITY GRID ***
   0.5797    0.5913    0.1837    0.4149    0.5861
```

$\boxed{0.5913}$ MAX.

Figure 3. --Defense Mode Deployment

1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.0														
0.0	1.00	0.0	1.00	1.00	0.0	1.00	1.00	1.00	0.0	1.00	0.0	1.00	1.00	0.0
1.00														
1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	0.0	1.00
0.0														
1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	0.0
2.00														
1.00	1.00	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	0.0	0.0	1.00
2.00														
0.0	1.00	0.0	0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.0	2.00
2.00														
1.00	1.00	0.0	0.0	1.00	1.00	0.0	1.00	1.00	1.00	0.0	1.00	1.00	1.00	2.00
1.00														
1.00	1.00	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	0.0	0.0	2.00
1.00														
0.0	1.00	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00
1.00														
1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.0	0.0	0.0	1.00	2.00
1.00														
0.0	1.00	1.00	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2.00														
0.0	0.0	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	1.00	1.00	1.00	2.00	2.00
1.00														
1.00	0.0	0.0	1.00	1.00	0.0	1.00	0.0	1.00	1.00	1.00	1.00	2.00	2.00	1.00
2.00														
1.00	0.0	1.00	1.00	1.00	1.00	0.0	1.00	1.00	1.00	0.0	1.00	2.0	1.00	2.00
1.00														
0.0	0.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	2.00
2.00														

Figure 4. -- The CSW Route Difficulty Grid

********** THE CSW ROUTE SELECTED **********

4124.5	4365.4
4074.5	4366.8
4025.9	4418.2
3974.6	4369.6
3924.6	4371.0
3876.0	4422.3
3827.4	4473.7
3778.8	4525.1
3727.4	4476.5
3677.5	4477.8
3627.5	4479.2
3576.1	4430.6
3526.1	4432.0
3476.2	4433.4
3427.6	4484.8
3376.2	4436.2
3324.8	4387.6
3276.2	4438.9
3226.2	4440.3
3176.3	4441.7
3126.3	4443.1
3076.3	4444.4
3026.3	4445.8
2975.0	4397.2
2925.0	4398.6
2875.0	4400.0

Figure 5.--The CSW Route Selected

unit is moved at a constant velocity between two specified plane departure points, and the movement event time is specified by input data.

If the dismounted CSW units are deploying in the defensive mode, each unit is moved along a straight line path from the dismount point to its desired defensive firing position. The unit will remain in its defensive firing position until it is determined that the units are to remount the APC.

If the CSW units are deployed in the attack mode, each unit moves along its selected route to its primary desired firing position. Once the unit attains its primary firing position, the unit's firing mission commences as determined by the firing models. "Shoot and scoot" tactics are performed between the primary and alternate firing positions as specified by input tactical doctrine. An example of dismounted CSW unit movement events from an actual simulation run is given in Figure 6.

Remounting of CSW Units

The logic to move dismounted CSW units to the APC and to remount the units on the APC was developed and programmed during the research period described by this report. Subroutine MOUNT is used to return the CSW units to the APC when it is determined that the units are to remount.

The computational procedures of Subroutine MOUNT follow:

1. If the CSW units are currently in the process of returning to the APC; i. e., if the remount time for the APC, TMNTD(LAPC), is greater than ECLOCK(ICE), the current clock time of the current element; go to step 4, otherwise go to step 2.

2. Determine the maximum travel time, TRVMNT, for the CSW units to return to the APC for all CSW units still alove by calls to subroutine CSTIME for each crew. *

3. Determine TMNTD(LAPC), the time at which all CSW units will be ready to remount the APC; i. e.,

$$TMNDT(LAPC) = ECLOCK(ICE) + TRVMNT.$$

---

*All firing assignments for the CSW units are deleted during remount procedures. The APC, however, may continue to have firing assignments until the final stages of remounting the crews.

```
       CURRENT APC                    LOCATION            CLOCK
            2                      2875.0,4400.0          0.20

              CREW              LOCATION          CLOCK          PR FIR POS
                3             2875.0,4400.0       60.20        4217.2,4362.9
                4             2875.0,4400.0       60.20        4096.9,4947.7
                5             2875.0,4400.0       60.20        3636.8,5624.9

        CURRENT ELEMENT     3 IS CSW CREW    1
CSMOVE1    1   1      30.00    2375.00    2100.00    2345.75    2140.55      0.0      0.0      50.00      5.00
CSMOVE3    2375.00    2100.00    2345.75    2140.55      50.00      30.00     50.00

CSMOVE4 CRATIO=    0.6000CDXY=      30.00THUTA=  2.19569

              CREW              LOCATION          CLOCK          PR FIR POS
                1             2357.4,2124.3       90.10        2345.7,2140.6

        CURRENT ELEMENT     4 IS CSW CREW    2
CSMOVE1    2   1      30.00    2375.00    2100.00    2433.50    2018.90      0.0      0.0     100.00      5.00
CSMOVE3    2375.00    2100.00    2433.50    2018.90     100.00      30.00    100.00

CSMOVE4 CRATIO=    0.3000CDXY=      30.00THUTA= -0.94591

              CREW              LOCATION          CLOCK          PR FIR POS
                2             2392.6,2075.7       90.10        2433.5,2018.9

        CURRENT ELEMENT     5 IS CSW CREW    3
CSMOVE1    3  25      30.00    2875.00    4400.00    2924.98    4398.62      0.0      0.0      50.00      5.00
CSMOVE3    2875.00    4400.00    2924.98    4398.62      50.00      30.00     50.00

CSMOVE4 CRATIO=    0.6000CDXY=      30.00THUTA= -0.02766

              CREW              LOCATION          CLOCK          PR FIR POS
                3             2905.0,4399.2       90.20        4217.2,4362.9

        CURRENT ELEMENT     6 IS CSW CREW    4
CSMOVE1    4  25      30.00    2875.00    4400.00    2920.63    4420.45      0.0      0.0      50.00      5.00
CSMOVE3    2875.00    4400.00    2920.63    4420.45      50.00      30.00     50.00

CSMOVE4 CRATIO=    0.6000CDXY=      30.00THUTA=  0.42135

              CREW              LOCATION          CLOCK          PR FIR POS
                4             2902.4,4412.3       90.20        4096.9,4947.7

        CURRENT ELEMENT     7 IS CSW CREW    5
CSMOVE1    5  28      30.00    2875.00    4400.00    2943.86    4416.05      0.0      0.0      70.71      5.00
CSMOVE3    2875.00    4400.00    2943.85    4416.05      70.71      30.00     70.71

CSMOVE4 CRATIO=    0.4243CDXY=      30.00THUTA=  0.22894
```

Figure 6.--Dismounted CSW Unit Movement Events

4. If all crews will reach the APC in the current event; i. e.,

$$TMNTD(LAPC) - ECLOCK(ICE) < EVTIME:$$

go to step 8, otherwise go to step 5.

5. Move each CSW unit which has not arrived at the remount position toward the APC by calls to subroutine CSMOVE.

6. If all of the crews are casualties, delete the APC's CSW carrier function for the duration of the battle and return to CSWCON, otherwise go to step 7.

7. Set the event time for the current event; i. e., TIME = EVTIME; allow the APC to fire while crews are returning to the APC, and return to CSWCON.

8. Set flags to indicate that all CSW units have remounted the APC in the current event; i. e., set (for each CSW unit, NCR),

$$KSMAKE(NCR) = 0$$
$$KSALT(NCR) = 0.$$

9. Set the current event time; allow the APC no further activities during the event; and return to subroutine CSWCON.

It should be noted that if the APC becomes a casualty and is no longer able to move, the dismounted CSW units assigned to that APC remain at their firing positions for the duration of the battle.

Summary and Conclusions

The ability of dismounted CSW units to attain covered and/or concealed positions is their primary asset in a land combat engagement. Thus, the primary goal of the crew-served weapon models is to provide a dynamic representation of dismounted CSW units' interactions with the terrain.

In particular, the selection of deployment modes, firing positions, and routes to these firing positions, as well as movement over these selected routes, emphasize the interaction of the dismounted CSW units with terrain conditions. In addition the models explicitly consider known enemy locations, CSW unit firepower capabilities, enemy threat, and input tactical doctrine.

Finally, it is important to note that the current models explicitly represent dismounted CSW units in support of armored units againt hard-point targets.

REFERENCES

1. Demmy, W. S., "Crew-Served Weapons," Chapter 9 (in) The Land Combat Model (DYNCOM), edited by A. B. Bishop and G. M. Clark, RF 2376 FR-1 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

2. Parry, S. H., "Crew-Served Weapon Movement Model," Chapter 5 (in) The Land Combat Model (DYNCOM), edited by Gordon M. Clark, RF 2376 FR-2 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, January 1, 1970, AD B001-826.

3. Parry, S. H., "Movement Model," Chapter 7 (in) The Tank Weapon System, edited by Albert B. Bishop and Gordon M. Clark, AR 69-2B (U), Systems Research Group, The Ohio State University, September 1969, AD 864920.

APPENDIX A

COMMON AREA DESCRIPTIONS

Commons Contained in this Appendix and Report
in which they were Previously Published

ADJASP	RF 3649 FR 74-1 (U)
AMBDA	"
AMOSPY	"
ANEUT	RF 2978 FR 71-3B (U)
ANGLIM	"
ANGSEC	RF 3649 FR 74-1 (U)
ARFCW1	"
ARFCW2	"
ARFCW3	"
ARFCW4	"
ARFCW5	"
ARFCW6	"
ARFCW7	"
ARFCW8	"
ARFCW9	"
ATDIR	RF 2978 FR 71-3B (U)
ATLIM	"
AVA	RF 3649 FR 74-1 (U)
BLUDET	"
BNEUT	RF 2978 FR 71-3B (U)
BRAIR	RF 3649 FR 74-1 (U)
CBDET	RF 2978 FR 71-3B (U)
CBERR	"
CBNEUT	RF 2978 FR 70-1 (U)
CCLOCK	RF 3649 FR 74-1 (U)
CF	RF 2978 FR 71-3B (U)
CFUEL	"
CKANG	"
CLKFB	"
COPAX	RF 3649 FR 74-1 (U)
COPAY	"
COPAZ	"
COPPX	"
COPPY	"
COPPZ	"
COPT	RF 2978 FR 71-3B(U)
COPVX	RF 3649 FR 74-1 (U)
COPVY	"
COPVZ	"
CSDSP	"
CSDEE	"
CSWDF1	"
CSWIND	"
CSXRT	"
CSYRT	RF 3649 FR 74-1 (U)
CW	"
CXCUR	"
CYCUR	"
DCW	"
DDPTS	RF 2376 FR 70-4A (U)
DEFFCN	RF 2978 FR 71-3B (U)
DELAY	"
DELD	"
DELR	"
DESAIR	RF 3649 FR 74-1 (U)
DISMTT	"
DTPTS	RF 2376 FR 70-4A (U)
DURC	RF 2978 FR 71-3B (U)
DURFB	RF 3649 FR 74-1 (U)
DURON	RF 2978 FR 71-3B (U)
DWC	RF 3649 FR 74-1 (U)
EC	"
ECLOCK	"
EDELX	RF 2978 FR 71-3B (U)
ELOCZ	"
EMICR	RF 2376 FR 70-4A (U)
ESMH	RF 2978 FR 71-3B (U)
EVHTIM	RF 3649 FR 74-1 (U)
EVTIM	RF 2978 FR 71-3B (U)
FACTL	RF 3649 FR 74-1 (U)
FMHITP	RF 2978 FR 71-3B (U)
FMOKIL	RF 3649 FR 74-1 (U)
FOCOM	"
FORMSZ	RF 2978 FR 71-3B (U)
FORMZS	"
GOEXAM	"
HALTDS	"
HALTDU	"
HCREW	"
HFORMS	"
HITP	"
HITPRB	"
HSPEED	"
HXRT	"
HYRT	"
HZRT	"
IANYPT	"
IBTAMO	RF 3649 FR 74-1 (U)

ICAP	RF 2978 FR 71-3B (U)	KSMAKE	RF 3649 FR 74-1 (U)
ICAP2	"	KSMUT	"
ICECOM	RF 3649 FR 74-1 (U)	KSWHCH	"
ICSWEL	"	KWAVAL	RF 2978 FR 71-3B (U)
ICSWGP	"	LAMMO	RF 3649 FR 74-1 (U)
IFBMIS	RF 2978 FR 71-3B (U)	LAPCCW	"
IFDCNT	"	LASTFB	"
IFMC	"	LCB	RF 2978 FR 71-3B (U)
IFRFL	"	LCREW	RF 2376 FR 70-4A (U)
IHAMO	RF 3649 FR 74-1 (U)	LCSWFN	"
IHDFMC	"	LCSWTP	RF 3649 FR 74-1 (U)
IHFNA	RF 2978 FR 71-3B (U)	LFLAG	RF 2978 FR 71-3B (U)
IHTARG	RF 3649 FR 74-1 (U)	LKILL	RF 2376 FR 70-4A (U)
INPRIR	RF 2978 FR 71-3B (U)	LMOVF	"
INTRIG	"	LMUFL	RF 2978 FR 71-3B (U)
IPHASE	"	LNSET	"
IREQ	"	LNUM	RF 2376 FR 70-4A (U)
IRFNH	"	LTHTNK	RF 3649 FR 74-1 (U)
ISACT	"	LWCOD	RF 2978 FR 71-3B (U)
ISCHPR	"	LWSYS	RF 3649 FR 74-1 (U)
ISFNA	"	LZZPE	RF 2978 FR 71-3B (U)
ITMASS	RF 3649 FR 74-1 (U)	MAINPR	RF 2376 FR 70-4A (U)
ITRIG	RF 2978 FR 71-3B (U)	MANACT	RF 2978 FR 71-3B (U)
ITYPA	RF 3649 FR 74-1 (U)	MAXWEP	RF 3649 FR 74-1 (U)
IUNACT	RF 2978 FR 71-3B (U)	MCLASS	RF 2978 FR 71-3B (U)
JAMOAV	RF 3649 FR 74-1 (U)	MD	RF 2376 FR 70-4A (U)
JPHASE	RF 2978 FR 71-3B (U)	MDFCSW	RF 3649 FR 74-1 (U)
JUNACT	"	MESARR	RF 2978 FR 71-3B (U)
KADV	"	MESCON	"
KAMAVL	RF 3649 FR 74-1 (U)	MFCSW	RF 3649 FR 74-1 (U)
KAMMAX	"	MIDATA	"
KAMOAV	"	MIFO	"
KAMPRD	"	MISAVE	RF 2978 FR 71-3B (U)
KANCEL	RF 2978 FR 71-3B (U)	MISHIT	"
KELAGN	RF 3649 FR 74-1 (U)	MISION	"
KFDC	RF 2978 FR 71-3B (U)	MNTRIG	"
KFO	"	MPTR	"
KFOD	RF 3649 FR 74-1 (U)	MREADY	"
KILFIR	RF 2978 FR 71-3B (U)	NAMO	RF 3649 FR 74-1 (U)
KLCPE	RF 3649 FR 74-1 (U)	NAVSEC	RF 2978 FR 71-3B (U)
KMANU	RF 2978 FR 71-3B (U)	NAXIS	"
KPATRN	"	NCBF	"
KPHAS	"	NCONC	"
KTUBE	"	NDELPT	"
KSALT	RF 3649 FR 74-1 (U)	NFB	"
KSDECN	"	NFLCRT	"
KSDSMT	"	NFMMAX	"

NFOCRT	RF 2978 FR 71-3B (U)
NFOFR	"
NFOMAX	"
NFRSAV	"
NFUCRT	"
NKPAT	"
NLNMAX	"
NM	"
NMLIM	"
NOBVH	"
NOUTFG	RF 3649 FR 74-1 (U)
NPHAS	RF 2978 FR 71-3B (U)
NRDCNT	RF 3649 FR 74-1 (U)
NREQR	RF 2978 FR 71-3B (U)
NRL	"
NRNDAT	RF 3649 FR 74-1 (U)
NRNDCN	RF 2978 FR 71-3B (U)
NSCHCN	"
NSTHFF	RF 3649 FR 74-1 (U)
NTELE	RF 2978 FR 71-3B (U)
NTOBAL	"
NTOT	RF 3649 FR 74-1 (U)
NTRANS	"
NUMFIR	RF 2978 FR 71-3B (U)
NUMSEN	RF 3649 FR 74-1 (U)
NUMBER	RF 2978 FR 71-3B (U)
NUMTRG	"
NXCONC	"
NXMIS	"
ONDIST	"
OPEN	"
ORESPN	"
OUTPST	"
PALLOW	"
PHNG	"
PKART	"
PKPB	"
PKTNK	"
PNG	"
RADINC	"
RADMA	"
RADMAX	"
RAFMNF	"
RAFMNL	"
RAM	"
RATT	"
RDFMIN	"
REACT	"
REDDET	RF 3649 FR 74-1 (U)
REMONT	"
RETFCN	RF 2978 FR 71-3B (U)
RF	"
RFOMAX	"
RFUEL	"
RGPTS	RF 2376 FR 70-4A (U)
RLNMAX	RF 2978 FR 71-3B (U)
RM	RF 2376 FR 70-4A (U)
RMAX	RF 3649 FR 74-1 (U)
RMIN	"
RSTAS	RF 2978 FR 71-3B (U)
RSTAU	"
RTKONH	RF 3649 FR 74-1 (U)
RTRIG	RF 2978 FR 71-3B (U)
RXFIRE	RF 3649 FR 74-1 (U)
RYFIRE	"
SATARG	RF 2978 FR 71-3B (U)
SC	RF 3649 FR 74-1 (U)
SCANNS	New
SCAPH	RF 2978 FR 71-3B (U)
SCHTIM	"
SDL	"
SFR	"
SHBM	RF 3649 FR 74-1 (U)
SIGADJ	"
SIGAL	"
SIGDEF	RF 2978 FR 71-3B (U)
SIGFFE	RF 3649 FR 74-1 (U)
SIGONP	"
SIGPBR	RF 2978 FR 71-3B (U)
SIGPLT	RF 3649 FR 74-1 (U)
SIGRAN	RF 2978 FR 71-3B (U)
SIGRBR	"
SIGSEN	"
SKY	"
SLSTIM	"
SPDPHS	"
SPDSE	"
SPEEDS	"
SPHAS	"
SPINC	"
SRPTS	RF 2376 FR 70-4A (U)
SSR	RF 2978 FR 71-3B (U)
STNEUT	RF 2978 FR 70-1 (U)
STRMIS	"
STRTIM	RF 3649 FR 74-1 (U)
STRTMS	RF 2978 FR 71-3B (U)

TARASP	RF 2978 FR 71-3B (U)
TARSPD	"
TBFR	"
TBL	"
TBSR	"
TC	RF 3649 FR 74-1 (U)
TCRIT	RF 2978 FR 71-3B (U)
TDFRDY	"
TDUD	"
TEC	RF 3649 FR 74-1 (U)
TEW	"
TF	RF 2376 FR 70-4A (U)
TFDCKL	RF 2978 FR 71-3B (U)
TFLY	"
THBM	RF 3649 FR 74-1 (U)
TIFRDY	RF 2978 FR 71-3B (U)
TIMBE	"
TINIT	"
TLOAD	"
TMISUN	"
TMNTD	RF 3649 FR 74-1 (U)
TNEUTM	"
TPMKH	RF 2978 FR 71-3B (U)
TRET	"
TRNET	RF 3649 FR 74-1 (U)
TVMIS	RF 2376 FR 70-4A (U)
TVOPT	"
TYPMIS	RF 3649 FR 74-1 (U)
UADJ	"
UAL	"
UFFE	"
UONP	"
UPLT	"
UPRBRA	RF 2978 FR 71-3B (U)
URNBRA	"
USEN	"
VEHSPD	"
VM	RF 3649 FR 74-1 (U)
VULRAD	RF 2978 FR 71-3B (U)
WAITAD	"
WFUEL	"
WPAT	"
WPHAS	"
WT	"
XCONC	"
XCSWIN	RF 3649 FR 74-1 (U)
XDF	"
XDFA	"
XFAC	RF 2978 FR 71-3B (U)
XFB	"
XLOC	"
XLSTAS	"
XLSTAU	"
XPHAS	RF 3649 FR 74-1 (U)
XS	RF 2978 FR 71-3B (U)
XSAVE	"
XTRIG	"
YCSWIN	RF 3649 FR 74-1 (U)
YDF	"
YDFA	"
YFAC	"
YFB	New
YLOC	RF 3649 FR 74-1 (U)
YPHAS	"
YSAVE	RF 2978 FR 71-3B (U)
ZM	RF 2376 FR 70-4A (U)
ZMD	"
ZRT	RF 3649 FR 74-1 (U)
ZSAVE	RF 2978 FR 71-3B (U)

COMMON/ADJASP/

CONTENTS: ADJASP contains the slant range adjustment factor for aerial target aspect.

ARRANGEMENT: ADJASP is a three-dimensional array.

where ADJASP(I, J, K)

$$I = \begin{cases} 1 \sim \text{incoming aspect} \\ 2 \sim \text{outgoing aspect} \end{cases}$$

$$J^1 = \begin{cases} 1 \sim \text{relative to firer} \\ 2 \sim \text{relative to MAD (see COMMON/MADRES/)} \end{cases}$$

K = firer weapon-ammunition code, (K = 1..., MIHTPB), where MIHTPB = largest firer weapon ammunition code index number assigned in COMMON/IHTPRB/.

INITIALIZATION: Required

USERS: ADTGSL

LENGTH: 2 * 2 * MIHTPB

[1] For J = 2, only I = 1 is assigned a weight.

COMMON/AMBDA/

CONTENTS: AMBDA contains the mean time between fire missions for an FO Bravo.

ARRANGEMENT: AMBDA(I, J) (real, seconds)

$$I = \begin{cases} 1 \ \text{regular FO Bravo} \\ 2 \ \text{special FO Bravo} \end{cases}$$

J = 1, . . . , MWART supporting fire weapon code

INITIALIZATION: Required

USERS: AFO

LENGTH: 2*MWART

COMMON/AMOSPY/

CONTENTS: AMOSPY contains the ratio of the current ammunition supply to the critical level, defined by common area JAMOAV, of an ammunition type for an aerial vehicle element.

ARRANGEMENT: AMOSPY is a two-dimensional array.

AMOSPY (I, J) contains the ratio for the I^{th} ammo type for the J^{th} aerial vehicle

where

I = 1,..., NAMMOS, where NAMMOS is the number of ammunition codes allowed per element.

J = 1,..., MAXNAV, where MAXNAV is the maximum number of aerial vehicle elements.

INITIALIZATION: None required

USERS: WASAIR, AIRAMO

LENGTH: NAMMOS*MAXNAV

COMMON/ANEUT/

CONTENTS: ANEUT contains one-half of the range dimension of the artillery neutralization ellipse for each fire pattern type.

ARRANGEMENT: ANEUT is a two-dimensional array.

ANEUT(I,J) contains the dimension for the I^{th} fire pattern of the J^{th} artillery weapon code for I = 1,...,5; J = 1,...,N where N is the maximum artillery unit weapon code (real, meters).

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH

LENGTH: 5 * N

COMMON/ANGLIM/

CONTENTS: ANGLIM contains one-half the total azimuth field-of-view angle for helicopter elements.

ARRANGEMENT: ANGLIM is a one-dimensional array.

ANGLIM(I) (real, radians)

where

I = 1,...,N[1] helicopter element weapon codes, and

N = total number of helicopter element weapon codes.

INITIALIZATION: Required

USERS: INTELL, ATAN5

LENGTH: N

---

[1]All elements in a helicopter section have identical characteristics. For element J in a section, I is computed by the relation

I = LWCOD(J) - MAXLWC.

See COMMON/LWCOD/ and COMMON/NUMBER/.

COMMON/ANGSEC/

CONTENTS: ANGSEC contains the sector of responsibility angle for all CSW units on an APC denoted by LAPC, measured from the APC coordinates (XC, YC). Used to compute firing positions for dismounted crews when attack is specified.

ARRANGEMENT: ANGSEC (LAPC) (real, radians)

where LAPC = 1, . . . , NAPCMX.

INITIALIZATION: Required

USERS: CSWCON

LENGTH: NAPCMX

COMMON/ARFCW1/

CONTENTS:	ARFCW1 contains a weighting factor for a heavy weapon being considered as a target for aerial vehicle fire.
ARRANGEMENT:	ARFCW1 is a one dimensional array.
	ARFCW1(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N
	where N is the maximum ground element weapon code.
INITIALIZATION:	Required
USERS:	AIRPOR
LENGTH:	N

COMMON/ARFCW2/

CONTENTS:	ARFCW2 contains a weighting factor for a light weapon being considered as a target for aerial vehicle fire.
ARRANGEMENT:	ARFCW2 is a one dimensional array.
	ARFCW2(I) contains the weighting factor for a target of weapon code I for I 1, . . . , N
	where N is the maximum ground element weapon code.
INITIALIZATION:	Required
USERS:	AIRPOR
LENGTH:	N

COMMON/ARFCW3/

CONTENTS:	ARFCW3 contains a weighting factor for an element which is being considered as a targer for aerial vehicle fire and which has fired at a friendly ground element in its previous event.
ARRANGEMENT:	ARFCW3 is a one dimensional array.
	ARFCW3(I) contains the weighting factor for a target of weapon code I for I - 1, . . . , N
	where N is the maximum ground element weapon code.
INITIALIZATION:	Required
USERS:	AIRPOR
LENGTH:	N

COMMON/ARFCW4/

CONTENTS:	ARFCW4 contains a weighting factor for an element which is being considered as a target for aerial vehicle fire and has fired at another friendly aerial vehicle in its previous event.
ARRANGEMENT:	ARFCW4 is a one dimensional array.
	ARFCW4(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N
	where N is the maximum ground element weapon code.
INITIALIZATION:	Required
USERS:	AIRPOR
LENGTH:	N

COMMON/ARFCW5/

CONTENTS: ARFCW5 contains a weighting factor for an element which is being considered as a target for aerial vehicle fire and which has fired at another friendly aerial vehicle of the current section in its last event.

ARRANGEMENT: ARFCW5 is a one dimensional array.

ARFCW5(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N

where N is the maximum ground element weapon code.

INITIALIZATION: Required

USERS: AIRPOR

LENGTH: N

COMMON/ARFCW6/

CONTENTS: ARFCW6 contains a weighting factor for an element which is being considered as a target for aerial vehicle fire which is presently being fired on by a friendly ground element.

ARRANGEMENT: ARFCW6 is a one dimensional array.

ARFCW6(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N

where N is the maximum ground element weapon code.

INITIALIZATION: Required

USERS: AIRPOR

LENGTH: N

COMMON/ARFCW7/

CONTENTS: ARFCW7 contains a weighting factor for an element which is being considered as a target for aerial vehicle fire which is presently being fired on by another firendly aerial vehicle.

ARRANGEMENT: ARFCW7 is a one dimensional array.

ARFCW7(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N

where N is the maximum ground element weapon code.

INITIALIZATION: Required

USERS: AIRPOR

LENGTH: N

COMMON/ARFCW8/

CONTENTS: ARFCW8 contains a weighting factor for an element which is being considered as a target for aerial vehicle fire which is presently being fired on by another aerial vehicle in the current section.

ARRANGEMENT: ARFCW8 is a one dimensional array.

ARFCW8(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N

where N is the maximum ground element weapon code.

INITIALIZATION: Required

USERS: AIRPOR

LENGTH: N

COMMON/ARFCW9/

CONTENTS: ARFCW9 contains the weighting factor for an element which is being considered as a target for aerial vehicle fire and which is currently being detected by another aerial vehicle in the current section.

ARRANGEMENT: ARFCW9 is a one dimensional array.

ARFCW9(I) contains the weighting factor for a target of weapon code I for I = 1, . . . , N

where N is the maximum ground element weapon code.

INITIALIZATION: Required

USERS: AIRPOR

LENGTH N

COMMON/ATDIR/

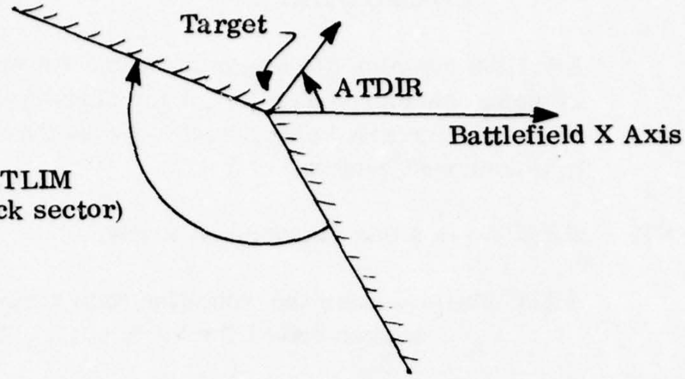
CONTENTS: ATDIR contains an array that specifies the orientation of a sector that is used by aerial sections conducting direct fire attacks. The sector is defined by the limits in COMMON/ATLIM/. See Figure.

ARRANGEMENT: ATDIR is a one-dimensional array.

ATDIR(I) whose entries are defined on the interval $(-\pi, \pi)$ (real, radians)

where

$$I = \begin{cases} 1 & \text{for the blue force, and} \\ 2 & \text{for the red force.} \end{cases}$$



INITIALIZATION: Required

USERS: RTATAK

LENGTH: 2

COMMON/ATLIM/

CONTENTS: ATLIM contains an array that specifies the angular limits of a sector from which direct fire attacks by aerial sections may be conducted. The orientation of the sector is given by the contents of COMMON/ATDIR/.

ARRANGEMENT: ATLIM is a one-dimensional array.

ATLIM(I) (real, radians)

where

$$I = \begin{cases} 1 & \text{for the blue force, and} \\ 2 & \text{for the red force.} \end{cases}$$

INITIALIZATION: Required

USERS: RTATAK

LENGTH: 2

COMMON/AVA/

CONTENTS: AVA contains presented areas of helicopter targets and is used in the ground-to-air detection model. All elements in a helicopter section have identical characteristics.

ARRANGEMENT: AVA is a two dimensional array.

AVA(I, J) (real, meters^2)

where

$$I = \begin{cases} 1 & \text{frontal cross sectional area} \\ 2 & \text{side cross sectional area} \\ 3 & \text{bottom cross sectional area,} \end{cases}$$

J = 1, . . . , NA3 helicopter weapon codes

where

NA3 = maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered for 1, . . . , MAXLWC for ground elements and from MAXLWC + 1, . . . , MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC, and for an arbitrary element K in a section, J is computed by the relation J = LWCOD(K) - MAXLWC.

INITIALIZATION: Required

USERS: DETH

LENGTH: 3*NA3

COMMON/BLUDET/

CONTENTS: BLUDET contains the detection codes for each blue element; a 0 for no knowledge, a 1 for approximate knowledge, a 2 for full visual detection, and a 3 for a pinpoint detection.

ARRANGEMENT: BLUDET is a two-dimensional array of halfwords:

BLUDET(I, J) INTEGER * 2

I = 1,...,NUMBLU the blue element numbers
J = 1,...,NUMELE - NUMBLU the red element numbers reduced to the range 1,...,NRED

where

NUMBLU is the total number of blue elements, NRED is the total number of red elements, and NUMELE is the total number of elements.

INITIALIZATION: None

USERS: AIRFIR, AIRPOR, GETDET, IDET, IDETP, KOMDET

LENGTH: NUMBLU * NRED halfwords = (NUMBLU * NRED), H

COMMON/BNEUT/

CONTENTS: BNEUT contains one-half the deflection dimension of an artillery neutralization ellipse for each fire pattern type.

ARRANGEMENT: BNEUT is a two-dimensional array.

BNEUT(I,J) contains the dimension for the I^{th} fire pattern of the J^{th} artillery weapon code for I = 1,...,5; J = 1,...,N where N is the maximum artillery unit weapon code (real, meters).

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH

LENGTH: 5 * N

COMMON/BRAIR/

CONTENTS: BRAIR contains firing data for an aerial vehicle of a given weapon code when firing a given suppressive fire weapon.

ARRANGEMENT: BRAIR is a three dimensional array.

BRAIR(I, J, K) contains the data for ammunition code I by aerial vehicle weapon codes

I = 1, . . . , 6 ammo types,

$$J = \begin{cases} 1 & \text{number of rounds fired per burst} \\ 2 & \text{desired number of bursts per firing event} \\ 3 & \text{firing rate in rounds per second} \\ 4 & \text{time between bursts} \end{cases}$$

K = 1,..., N, where N is the maximum number of aerial vehicle weapon codes.

INITIALIZATION: None

USERS: HFIRE, WASAIR

LENGTH: 6*4*N

COMMON/CBDET/

CONTENTS: CBDET contains the rate at which an aerial unit detects an artillery unit to be observed during a counterbattery mission.

ARRANGEMENT: CBDET is a one-dimensional array.

CBDET(I) (real, seconds^{-1})

where

I = 1,...,MWAIR-MWMIS aerial unit weapon codes.[1]

INITIALIZATION: Required

USERS: CBOBS

LENGTH: MWAIR - MWMIS

[1]See COMMON/CBERR/ for a definition of aerial unit weapon code.

COMMON/CBERR/

CONTENTS: CBERR contains parameters of the exponential function that specifies the standard deviation of the distribution of errors in battery location reported by an aerial unit that has detected a battery during a counterbattery observation mission.

ARRANGEMENT: CBERR is a two-dimensional array.

CBERR(I,J) (real)

where

$$I = \begin{cases} 1 & \text{indicates the value of the function after} \\ & \text{zero observation time (meters)} \\ 2 & \text{indicates the decay rate (meters/second)} \end{cases}$$

J = 1,...,MWAIR-MWMIS aerial unit weapon codes.[1]

INITIALIZATION: Required

USERS: CBCONT

LENGTH: 2 * (MWAIR-MWMIS)

---

[1]For aerial unit N, the aerial unit weapon code is

J = LWCOD(NUMELE+NUMART+MISTUN+N) - MWMIS.

See COMMON/NUMBER/ and COMMON/LWCOD/.

COMMON/CBNEUT/

CONTENTS: CBNEUT contains radii about firing batteries in which incoming projectiles produce slight neutralization to the entire battery operation (see COMMON/STNEUT/).

ARRANGEMENT: CBNEUT(I) contains the radius about battery I (real, meters)

for I = 1, ..., N

where

N is the number of firing batteries.

INITIALIZATION: Required.

USERS: ARLETH

LENGTH: N

COMMON/CCLOCK/

CONTENTS: CCLOCK contains the current clock time of CSW teams.

ARRANGEMENT: CCLOCK is a one-dimensional array.

CCLOCK(I) (real, seconds)

I = 1,..., NCSWMX crew numbers

INITIALIZATION: None

USERS: CSMOVE, CSWCON

LENGTH: NCSWMX

COMMON/CF/

CONTENTS: CF stores an array containing the critical threat ratio used in determining if an intense fire fight exists (real).

ARRANGEMENT: CF is a one-dimensional array.

CF(K) the ratio of known enemy firers to friendly survivors which, if exceeded, determines that an intense fire fight exists for an FO of type K where

$$K = \begin{cases} 1 & \text{blue artillery FO,} \\ 2 & \text{red artillery FO,} \\ 3 & \text{blue MISTIC FO,} \\ 4 & \text{red MISTIC FO,} \\ 5 & \text{blue special FO, and} \\ 6 & \text{red special FO.} \end{cases}$$

INITIALIZATION: Required

USERS: ARFO, DARFO, HELFIR

LENGTH: 6

COMMON/CFUEL/

CONTENTS: CFUEL contains the critical fuel quantity for each element in an aerial section.

ARRANGEMENT: CFUEL is a one-dimensional array.

CFUEL(I) (real, pounds)

where

I = 1,...,N helicopter weapon codes (see COMMON/ANGLIM/ for helicopter weapon code definition).

INITIALIZATION: Required

USERS: RETIRE, FUELD

LENGTH: N

COMMON/CKANG/

CONTENTS: CKANG represents one-half limiting angle of aerial observer's scanning pattern.

ARRANGEMENT: CKANG is a one-dimensional array.

CKANG(I) (real, radians)

where

$$I = \begin{cases} 1 & \text{wide angle search used while looking for targets} \\ 2 & \text{narrow angle search used while element has a target.} \end{cases}$$

INITIALIZATION: Required

USERS: DETT, ATAN5

LENGTH: 2

COMMON/CLKFB/

CONTENTS: CLKFB contains the battle time of the previous event for each artillery firing battery.

ARRANGEMENT: CLKFB is a one-dimensional array.

CLKFB(I) (real, seconds)

where

CLKFB(I) contains the time for the I^{th} battery

for I = 1,...,N where N is the maximum number of artillery firing batteries.

INITIALIZATION: None

USERS: AFB, WRTART

LENGTH: N

COMMON/COPAX/

CONTENTS: COPAX stores an ordered array containing the X component of acceleration recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPX/, and COMMON /COPVX/).

ARRANGEMENT: COPAX is a two-dimensional array.

COPAX(K, N) (real, meters/second^2)

where

K = 1, . . . , RTSMAX movement data set; i. e., the number of instantaneous time, position, velocity, and acceleration component values to be stored for later interpolation (integer, set in APPINT, currently 20), and

N = 1, . . . , NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPAY/

CONTENTS: COPAY stores an ordered array containing the Y component of acceleration recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPY/, and COMMON /COPVY/).

ARRANGEMENT: COPAY is a two-dimensional array.

COPAY(K, N) (real, meters/second^2)

where

K = 1, . . . , RTSMAX movement data set, and

N = 1, . . . , NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPAZ/

CONTENTS: COPAZ stores an ordered array containing the Z component of acceleration recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPZ/, and COMMON /COPVZ/).

ARRANGEMENT: COPAZ is a two-dimensional array.

COPAZ(K, N) (real, meters/second^2)

where

K = 1, . . . , RTSMAX movement data set, and

N = 1, . . . , NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPPX/

CONTENTS: COPPX stores an ordered array containing the battlefield X coordinates that specify the movement traces for aerial vehicle sections (see COMMON/COPT/, COMMON/COPVX/, and COMMON/COPAX/).

ARRANGEMENT: COPPX is a two-dimensional array.

COPPX(K, N) (real, meters)

where

K = 1, . . . , RTSMAX movement data sets; i.e., the number of instantaneous time, position, velocity, and acceleration component values to be stored for later interpolation (integer, set in APPINT, currently 20), and

N = 1, . . . NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPPY/

CONTENTS: COPPY stores an ordered array containing the battlefield Y coordinates that specify the movement traces for aerial vehicle sections (see COMMON/COPT/, COMMON/COPVY/, and COMMON/COPAY/).

ARRANGEMENT: COPPY is a two-dimensional array.

COPPY(K, N) (real, meters)

where

K = 1, . . . , RTSMAX movement data sets, and

N = 1, . . . , NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPPZ/

CONTENTS: COPPZ stores an ordered array containing the battlefield Z coordinates that specify the movement traces for aerial vehicle sections (see COMMON/COPT/, COMMON/COPVZ/, and COMMON/COPAZ/).

ARRANGEMENT: COPPZ is a two-dimensional array.

COPPZ(K, N) (real, meters)

where

K = 1, . . . , RTSMAX movement data sets, and

N = 1, . . . , NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPT/

CONTENTS: COPT stores an array containing clock times at which movement data sets for aerial vehicle sections are stored (see COMMON/COPPX/, COMMON/COPVX/ and COMMON/COPAX/).

ARRANGEMENT: COPT is a two-dimensional array.

COPT(K,N) (real, seconds)

where

K = 1,...,RTSMAX movement data sets, and

N = 1,...,NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX * NSEC

COMMON/COPVX/

CONTENTS: COPVX stores an ordered array containing the X components of velocity recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPX/, and COMMON /COPAX/).

ARRANGEMENT: COPVX is a two-dimensional array.

COPVX(K, N) (real, meters/second)

where

K = 1,..., RTSMAX movement data sets; i.e., the number of instantaneous time, position, velocity, and acceleration component values to be stored for later interpolation (integer, set in APPINT, currently 20), and

N = 1,..., NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPVY/

CONTENTS: COPVY stores an ordered array containing the Y component of velocity recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPY, and COMMON /COPAY/).

ARRANGEMENT: COPVY is a two-dimensional array.

COPVY(K, N) (real, meters/second)

where

K = 1,..., RTSMAX movement data sets, and

N = 1,..., NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/COPVZ/

CONTENTS: COPVZ stores an ordered array containing the Z component of velocity recorded for aerial vehicle sections (see COMMON/COPT/, COMMON/COPPZ/, and COMMON /COPAZ/).

ARRANGEMENT: COPVZ is a two-dimensional array.

COPVZ(K, N) (real, meters/second)

where

K = 1,..., RTSMAX movement data sets, and

N = 1,..., NSEC aerial section numbers.

INITIALIZATION: None

USERS: APPINT

LENGTH: RTSMAX*NSEC

COMMON/CSDSP/

CONTENTS: Common CSDSP contains the increase in tactical difficulty at a route-selection grid point due to an intervisible strong point at distance CDIST where TEC(K + 1) ≤ CDIST ≤ TEC(K). See COMMON/TEC/.

ARRANGEMENT: CSDSP is a one-dimensional array.

CSDSP(I) (real)

I = 1,..., NRWP + NBWP

where

NRWP is the number of red weapons at a strong point and NBWP is the number of blue weapons at a strong point.

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: NRWP + NBWP

COMMON/CSDEE/

CONTENTS: CSDEE contains the increase in tactical difficulty at a route selection grid point to a known intervisible enemy element at distance CEDIS, where TEC(K + 1) ≤ CEDIS ≤ TEC(K). See COMMON/TEC/.

ARRANGEMENT: CSDEE is a one-dimensional array.

CSDEE(I) (real)

I = 1,..., MNFRT, the total number of element weapon codes

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: MNFRT

COMMON/CSWDF1

CONTENTS: CSWDF1 contains event variables used by the CSW model.

ARRANGEMENT:
1) NCR = CSW unit number
2) NKWEP = weapon code of CSW unit
3) NCTOT = weapon code of CSW unit
4) LAST = the number of the last movement control point
5) JC = number of grid spacings to either side of the center of the route selection grid.
6) XC = current elements X-coordinate
7) YC = current elements Y-coordinate
8) LAPC = carrier's APC number
9) ICREW = CSW units regular element number

INITIALIZATION: None

USERS: CSWCON, CSMOVE, CSWGRD, CSRTSL, CRTLOC, CSTIME, CSWDES, CXYLOC, MOUNT

LENGTH: 9

COMMON/CSWIND/

CONTENTS: Common CSWIND contains general fixed variables used in the crew-served weapon (CSW) model.

ARRANGEMENT:
1) CALLOW - maximum allowable distance from primary to alternate desired firing position, expressed in units of SPACE.
2) ICRMAX - number of points in range-fire power index versus speed determination.
3) INCMAX - number of points for limiting speed determination.
4) KSP(1) - number of red strongpoints
5) KSP(2) - number of blue strongpoints
6) NAPCMX - total number of APC's in the battle
7) NCSWMX - total number of CSW teams used in the battle.
8) NCWMAX - maximum number of weapon codes used by crew served weapon teams.
9) RLIMIT - maximum length of line along which CSW teams are to be deployed in the defensive mode.
10) SPACE - distance between grid points of both the fire position grid and route selection grid in CSW model.
11) CEVTIM - current event time
12) MWC(1) - maximum number of red weapons at a red strongpoint.
13) MWC(2) - maximum number of blue weapons at a blue strongpoint.

INITIALIZATION: Required

USERS: CSWCON, CSMOVE, CSWGRD, CSRTSL, CSTIME, CRTLOC, CSWDES, CXYLOC

LENGTH: 13

COMMON/CSXRT/

CONTENTS:	CSXRT contains the X-coordinates of the I^{th} movement control point specifying the movement path of dismounted CSW unit NCR.
ARRANGEMENT:	CSXRT is a two-dimensional array.
	CSXRT(I, NCR) (real, meters)
	where
	I = 1,..., NPTMAX, (the maximum number of movement control points)
	NCR = 1,..., NCSWMX (CSW number)
INITIALIZATION:	None
USERS:	CSMOV, CSRTSL, CSWCON, CSWGRD, MOUNT
LENGTH:	NCSWMX*NPTMAX

COMMON/CSYRT/

CONTENTS:	CSYRT contains the Y coordinate of the I^{th} movement control point specifying the movement path of dismounted CSW unit NCR.
ARRANGEMENT:	CSYRT is a two-dimensional array.
	CSYRT(I, NCR) (real, meters)
	where
	I = 1,..., NPTMAX, the maximum number of movement control points
	NCR = 1,..., NCSWMX, CSW number
INITIALIZATION:	None
USERS:	CSMOV, CSRTSL, CSWCON, CSWGRD, MOUNT
LENGTH:	NCSWMX*NPTMAX

COMMON/CW/

CONTENTS: CW(I) – weight (relative importance) assigned to factor I in the attack fire-position-selection grid

ARRANGEMENT: CW(I) (real)

I = 1, weight of TGCOV (target cover)
I = 2, weight of CSWCO (CSW cover)
I = 3, weight of RFI (range-firepower index)

Note: $0 \leq CW(I) \leq 1$ and $\sum_{I=1}^{3} CW(I) = 1$

INITIALIZATION: Required

USERS: CSWDES

LENGTH: 3

COMMON/CXCUR/

CONTENTS: CXCUR contains the X-coordinate of a CSW current position

ARRANGEMENT: CXCUR is a one-dimensional array

CXCUR(I) (real, meters)

where

I = 1, NCSWMX, the maximum number of CSW's in the battle

INITIALIZATION: None

USERS: CSMOV, CSWCON, CSWGRD, MOUNT

LENGTH: NCSWMX

COMMON/CYCUR/

CONTENTS: CYCUR contains the Y coordinate of a CSW current position

ARRANGEMENT: CYCUR is a one-dimensional array

CYCUR(I) (real, meters)

where

I = 1, . . . , NCSWMX, the maximum number of CSW's in the battle

INITIALIZATION: None

USERS: CSMOV, CSWCON, CSWGRD, MOUNT

LENGTH: NCSWMX

COMMON/DCW/

CONTENTS: Common DCW contains the weight (relative importance) assigned to factor I in the defensive fire position selection grid.

ARRANGEMENT: DCW(I) (real)

where

I = 1 weight assigned relative to percent of threat location covered

I = 2 weight assigned relative to crew concealment

INITIALIZATION: Required

USERS: CSWGRD

LENGTH: 2

COMMON/DDPTS/

CONTENTS: DDPTS contains the flight-range (meters) curve for direct-fire missile flights.

ARRANGEMENT: DDPTS is a one-dimensional array.

DDPTS(I)

I = 1, 2, ···, N

where

DDPTS(I) = the distance down range at time DTPTS(I) (see COMMON/DTPTS),

DDPTS(1) = 0.0, and

N = number of distance points.

INITIALIZATION: Required

USERS: FINAL

LENGTH: N

COMMON/DEFFCN/

CONTENTS: DEFFCN contains the fraction of the nonretired portion of an aerial unit which must be seeking defensive positions before the unit itself will seek a defensive position.

ARRANGEMENT: DEFFCN is a one-dimensional array.

DEFFCN(I) (real)

where

I = 1,2,...,MWAIR-MWMIS aerial unit weapon codes.

INITIALIZATION: Required

USERS: ATDEC

LENGTH: MWAIR-MWMIS

COMMON/DELAY/

CONTENTS: DELAY contains coordinates of holding positions for aerial units. The positions are used when a unit wishes to retire, seek a defensive position or wait for mission assignment.

ARRANGEMENT: DELAY is composed of a three-dimensional array and four variables indicating the size of the array.

1. NBDP total number of blue holding positions
2. NRDP total number of red holding positions
3. NBRP number of blue retirement positions ($1 \le NBRP \le NBDP$)
4. NRRP number of red retirement positions ($1 \le NRRP \le NBDP$)
5. DELAY(I,J,K) (real, meters)

where

$$I = \begin{cases} 1 & \text{for blue force, and} \\ 2 & \text{for red force.} \end{cases}$$

$$J = \begin{cases} 1 & \text{for the holding position X coordinate, and} \\ 2 & \text{for the Y coordinate.} \end{cases}$$

$$K = \begin{cases} 1,\ldots,NBRP \text{ or } 1,\ldots,NRRP & \text{indicating the number of the retirement position, and} \\ NBRP+1,\ldots,NBDP+1 \text{ or } NRRP+1,\ldots,NRDP+1 & \text{indicating the number of the defensive or waiting position (values for } K = NBDP+1 \text{ and } K = NRDP+1 \text{ are reserved for a unit's present position).} \end{cases}$$

INITIALIZATION: Required for all entries except I = 1, 2; J = 1, 2; K = NBDP + 1 or K = NRDP + 1.

USERS: DEFPOS

LENGTH: 4 + 2 * 2 * (maximum of NBDP + 1 and NRDP + 1)

COMMON/DELD/

CONTENTS: DELD contains the deflection aim point adjustment dimension for an artillery firing battery in firing successive volleys of a walking fire pattern.

ARRANGEMENT: DELD is a two-dimensional array.

DELD(I,J) contains the dimension for the I^{th} volley of the J^{th} artillery unit weapon code for $I = 1,\ldots,6$; $J = 1,\ldots,N$ where N is the maximum artillery unit weapon code (real, meters).

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH

LENGTH: 6 * N

COMMON/DELR/

CONTENTS: DELR contains the range aim point adjustment dimension for an artillery firing battery in firing successive volleys of a walking fire pattern.

ARRANGEMENT: DELR is a two-dimensional array.

DELR(I,J) contains the dimension for the I^{th} volley of the J^{th} artillery unit weapon code for $I = 1,\ldots,6$; $J = 1,\ldots,N$ where N is the maximum artillery unit weapon code. (real, meters)

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH

LENGTH: 6 * N

CONTENTS: DESAIR contains desirability factors for an air section (consisting of a given number of suppressive and point fire weapons) engaging an enemy ground complex consisting of a given number of light and heavy targets.

ARRANGEMENT: DESAIR is a four-dimensional array. (Integer*2)

$$\text{DESAIR(I, J, K, L)} = \begin{cases} -1 & \text{nonallowed combination, or} \\ R & \text{desirability factor for a section firing against I-1 heavy targets, J-1 light targets, with K-1 suppressive fire weapons and J-1 point fire weapons} \end{cases}$$

where

R > 0

I = 1,..., NAVY + 1 where NAVY is the maximum number of heavy targets which may be engaged by an air section

J = 1,..., NLGT + 1 where NLGT is the maximum number of light targets which may be engaged by an air section

K = 1,..., NAFW + 1 where NAFW is the maximum number of suppressive fire weapons which may be employed by an air section

L = 1,..., NPFW + 1 where NPFW is the maximum number of point fire weapons which may be employed by an air section.

INITIALIZATION: Required

USERS: AIRFIR

LENGTH: ((NAVY + 1)*(NLGT + 1)*(NAFW + 1)*(NPFW +1)) , H

COMMON/DISMTT/

CONTENTS: Common DISMTT contains the time for a CSW team of weapon code, NKWEP, to dismount from their APC.

ARRANGEMENT: DISMTT(NKWP) (real, seconds)
NKWEP = 1, . . . , NCWMAX CSW weapon code

INITIALIZATION: Required

USERS: CSWCON, MOUNT

LENGTH: NCWMAX

COMMON/DTPTS/

CONTENTS: DTPTS contains the time (seconds) from launch for direct-fire missile flights.

ARRANGEMENT: DTPTS is a one-dimensional array.

DTPTS(I),

$I = 1, \cdots, N$

where DTPTS(I) is the time from launch at distance point I,

DTPTS(1) = 0.0, and

N = number of distance points.

INITIALIZATION: Required

USERS: FINAL

LENGTH: N

COMMON/DURC/

CONTENTS: DURC stores an array containing the estimated duration time of an FO Bravo target.

ARRANGEMENT: DURC(K,L) (real, seconds)

where

$$K = \begin{cases} 1 & \text{regular FO Bravo, and} \\ 2 & \text{special FO Bravo.} \end{cases}$$

L = 1,...,MWART artillery unit weapon code

INITIALIZATION: Required

USERS: AFO

LENGTH: 2 * MWART

COMMON/DURFB/

CONTENTS: DURFB contains the specified duration time of an artillery fire mission or the duration time of a target assigned to an aerial unit. Time durations for lauchers are not specified in DURFB.

ARRANGEMENT: DURFB is a one-dimensional array:

DURFB(I) (real, seconds)

contains the duration time for the I^{th} fire support firer

where

I = 1,...,NUMART + ITOTLN + NUMAVT firer support firer numbers

and

NUMART = number of artillery firing batteries

ITOTLN = number of launchers, and

NUMAVT = number of aerial vehicle teams.

Values of DURFB(I), I = NUMART + 1,..., NUMART + ITOTLN, are not currently used.

INITIALIZATION: None

USERS: AFB, AIRFB, MISEND, PRMSET, CMMIS

LENGTH: NUMART + ITOTLN + NUMAVT

COMMON/DURON/

CONTENTS: DURON stores an array containing the expected target duration time for an on-call target triggered by friendly elements.

ARRANGEMENT: DURON(I) (real, seconds)

where

I = 1,...,20 concentration number.

INITIALIZATION: Required

USERS: AFO, FSCHEK

LENGTH: 20

COMMON/DWC/

CONTENTS: Common DWC contains the weight (relative importance) assigned to factor I in the CSW route selection grid.

ARRANGEMENT: DWC(I) (real)

where

I = 1 weight of difficulty factors

I = 2 weight of travel time factors

INITIALIZATION: Required

USERS: CSTCDF

LENGTH: 2

COMMON/EC/

CONTENTS:	Common EC is the tactical difficulty grid used in CSW route selection procedures. This is a working area matrix.
ARRANGEMENT:	EC is a two-dimensional array.
	EC(I, J) (real)
	I = 1,..., 20 maximum grid points between objective and APC
	J = 1,..., 16 maximum grid width of route selection
INITIALIZATION:	Zero, altered during execution
USERS:	CSRTSL, CSTCDF
LENGTH:	20*16

COMMON/ECLOCK/

CONTENTS: ECLOCK contains the elapsed battle time for all elements. See Table 1.2, pgs. 1-16, RF2978-FR71-3A(U).

ARRANGEMENT: ECLOCK(I) (real, seconds)

where

I = 1,..., NUMELE subscripts giving the time for regular elements
I = N + 1,..., NSCLK subscripts giving the time for special clocks
I = NSCLK + 1,..., NMIS subscripts giving the time for missiles
I = NMIS + 1,..., NTFDC subscripts giving the time for forward observers
I = NTFO + 1,..., NTFDC subscripts giving the time for fire direction centers and radio nets
I = NTFDC + 1,..., NTFB subscripts giving the time for fire support firers
I = NTFB + 1,..., NCLK subscripts giving the time for artillery intelligence centers

and

NUMELE	=	number of regular elements
NSCLK	=	NUMELE + number of special clocks
NMIS	=	NSCLK + number of missiles
NTFO	=	NMIS + number of forward observers
NTFDC	=	NTFO + number of fire direction centers and radio nets
NTFB	=	NTFDC + number of fire support firers
NCLK	=	NTFB + number of artillery intelligence centers.

INITIALIZATION: Zero, altered during execution.

USERS:

ADGCON	ARLETH	CSWCAS	GETICE	MOUNT
ADTIC	ATKPRM	CSWCON	HGET	MOV
AFB	BTLETH	FIRCON	HRAPUP	MVCON
AFDC	CASELE	FIRKIL	INTELL	NEWFO
AFO	CBCONT	FIRMOD	INTERP	PASMIS
AFSC	CBOBS	FIXDAT	LAUNCH	PLOT
AIC	COM	FLIGHT	LMUSET	PREILL
AIRFB	CREATM	FSCHEK	MAIN	SEQCNT
ARFO	CSRTSL	GETHEL	MFB	SETSEQ
APFDYS	CSMOVE	FSCMON	MANOUT	SETLOS

USERS: (cont.)

SPOT
SPOTI
STACLK
TACMES
WRITFO
WRTART
WRTFRL
ZILCH

LENGTH: NCLK

COMMON/EDELX/

CONTENTS: EDELX contains formation spacing information for ground vehicles.

ARRANGEMENT: EDELX is a one-dimensional array.

EDELX(I) (real, meters)

where

I = 1,...,N subscript giving the formation lateral offset of the I^{th} element where N is the maximum number of elements. EDELX is measured relative to the formation leader and is positive if the position is to the right of the leader.

INITIALIZATION: Zero, altered during execution.

USERS: MOV

LENGTH: N

COMMON/ELOCZ/

CONTENTS: ELOCZ contains vertical position coordinates for helicopter elements measured relative to the local terrain elevation.

ARRANGEMENT: ELOCZ is a one-dimensional array.

ELOCZ(I) (real, meters)

where

I = 1,...,N helicopter element numbers.

INITIALIZATION: None

USERS: SECSET, RTSRCH, RTCROS, HELMOV

LENGTH: N

COMMON/EMICR/

CONTENTS: EMICR contains the micro-terrain deviation from the macro-terrain for each element.

ARRANGEMENT: EMICR is a linear array.

EMICR(I) (real, meters)

I = 1, ···, N element numbers

where N is the total number of elements.

INITIALIZATION: Required

USERS: GETICE, COV, FIRCON, RTSEL, MOV, CONC, LOSPC

LENGTH: N

COMMON/ESMH/

CONTENTS: ESMH corresponds exactly to ESM but is associated with estimates of difficulty prepared by a helicopter section leader.

ARRANGEMENT: ESMH(I) (real)

I = 1,...,N weapon code numbers of enemy elements

where

N = total number of weapon codes.

INITIALIZATION: Required

USERS: RTSELH

LENGTH: N

COMMON/EVHTIM/

CONTENTS: EVHTIM contains the standard event time for helicopter elements.

ARRANGEMENT: EVHTIM is a one-dimensional array.

EVHTIM(I) (real, seconds)

where

I = 1,...,NA3, where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,...,MAXLWC for ground elements and from MAXLWC + 1,...,MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC

INITIALIZATION: Required

USERS: ATKPRM, HELMOV, RTATAK, RTJOIN

LENGTH: NA3

COMMON/EVTIM/

CONTENTS: EVTIM contains the standard event time for a fire support element.

ARRANGEMENT: EVTIM is a one-dimensional array.

EVTIM(I) (real, seconds)

where

I = 1,...,MWAIR fire support weapon codes.

INITIALIZATION: Required

USERS: AIRFB, MFB, LAUNCH

LENGTH: MWAIR

COMMON/FACTL/

CONTENTS: FACTL contains firing data for aerial vehicles firing suppressive fire bursts prior to firing a point weapon.

ARRANGEMENT: FACTL is a two-dimensional array.

FACTL(I,J) contains the data for the J^{th} aerial vehicle

where

$$I = \begin{cases} 1 & \text{number of rounds per burst} \\ 2 & \text{number of burst to be fired} \\ 3 & \text{time per burst} \\ 4 & \text{time between bursts} \\ 5 & \text{total time to fire all bursts} \end{cases}$$

for J = 1,...,MAXNAV where MAXNAV is the maximum number of aerial vehicles.

INITIALIZATION: None

USERS: RTATAK, WASAIR

LENGTH: 5*MAXNAV

COMMON/FMHITP/

CONTENTS: FMHITP contains additional horizontal and vertical dispersions to be included in hit probability determination for a moving firer.

ARRANGEMENT: FMHITP is a four-dimensional array.

FMHITP(I,J,K,L) (real, mils)

where

I = 1,...,6 = horizontal dispersions corresponding to range intervals from COMMON/HPRNG/

I = 7,...,12 = vertical dispersions corresponding to range intervals from COMMON/HPRNG/

J = 1,...,6 = corresponds to the rough terrain code of the firer's location

K = 1,...,6 = corresponds to speed intervals as defined in COMMON/VEHSPD/

L = 1,...,M = corresponds to mobility characteristics where M is the maximum number of characteristics.

INITIALIZATION: Required

USERS: DDSX2

LENGTH: 12*6*6*M

COMMON/FMOKIL/

CONTENTS: FMOKIL contains the seconds of flight time remaining to a helicopter after damage is inflicted by an air defense weapon.

ARRANGEMENT: FMOKIL is a two-dimensional array.

FMOKIL(I,J) (real, seconds)

where

I = 1,..., NLETHC indicates the lethality code of the damage inflicted by an air defense weapon.

J = 1,..., NA3 indicates the weapon code of the damaged helicopter

where

NLETHC is the maximum number of lethality codes for an aerial vehicle target ($1 \leq NLETHC \leq 9$) and

NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,..., NAXLWC for ground elements and from MAXLWC + 1,..., MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: CASHEL

LENGTH: NLETHC*NA3

COMMON/FOCOM/

CONTENTS: FOCOM contains the time for an artillery firing battery to communicate the completion or deletion of a requested mission to the requesting forward observer.

ARRANGEMENT: FOCOM is a one-dimensional array.

I = 1,...,MWART

FOCOM(I) contains the time for the I^{th} artillery unit weapon code (real, seconds)

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: MWART

COMMON/FORMSZ/

CONTENTS: Z analog of FORMSX and FORMSY. Used only for aerial vehicles.

ARRANGEMENT: See COMMON/FORMSX/.

INITIALIZATION: Required

USERS: OFFSET

LENGTH: 7 * N

COMMON/FORMZS/

CONTENTS: Z analog of FORMXS and FORMYS. Used only for aerial vehicles.

ARRANGEMENT: See COMMON/FORMXS/.

INITIALIZATION: Required

USERS: OFFSET

LENGTH: 3

COMMON/GOEXAM/

CONTENTS: GOEXAM contains the time increment which is assessed to a ground-to-air communicator's clocktime when the communicator finds no search and destroy targets for aerial attack.

ARRANGEMENT: GOEXAM(I) (real, seconds)

where

$$I = \begin{cases} 1 & \text{blue force communicator, and} \\ 2 & \text{red force communicator.} \end{cases}$$

INITIALIZATION: Required

USERS: AFSC

LENGTH: 2

COMMON/HALTDS/

CONTENTS: HALTDS contains the desired altitude above the terrain for an aerial section.

ARRANGEMENT: HALTDS is a two-dimensional array.

HALTDS(I,J) (real, meters)

where

$$I = \begin{cases} 1 & \text{indicates section enroute movement} \\ 2 & \text{indicates section search} \\ 3 & \text{indicates section loiter} \\ 4 & \text{indicates the initial point of a direct-fire attack} \\ 5 & \text{indicates the launch altitude for indirect fire} \\ 6 & \text{indicates the initial point of illumination.} \end{cases}$$

J = 1,...,N aerial section weapon codes (see COMMON/ANGLIM/ for weapon code definition).

INITIALIZATION: Required

USERS: RTCROS, RTSRCH, RTLOIT, RTATAK

LENGTH: 6 * N

COMMON/HALTDU/

CONTENTS: HALTDU contains the desired altitude above the terrain for an aerial unit.

ARRANGEMENT: HALTDU is a two-dimensional array.

HALTDU(I,J) (real, meters)

where

$$I = \begin{cases} 1 \text{ indicates unit enroute movement} \\ 2 \text{ indicates unit search} \\ 3 \text{ indicates unit loiter.} \end{cases}$$

J = 1,...,N aerial unit weapon codes (see COMMON/CBERR/ for definition of weapon code).

INITIALIZATION: Required

USERS: RTCROS, RTSRCH, RTLOIT

LENGTH: 3 * N

COMMON/HCREW/

CONTENTS: HCREW contains the height of a dismounted crew-served weapon crew.

ARRANGEMENT: HCREW is a one-dimensional array.

HCREW(I)

where

I = 1,...,NCWMAX the maximum number of crew-served weapon codes.

INITIALIZATION: Required

USERS: CSRTSL, CSWDES, CSWGRD

LENGTH: NCWMAX

COMMON/HFORMS/

CONTENTS:	HFORMS contains the formation pattern number to be used by an aerial section while operating independently.
ARRANGEMENT:	HFORMS is a one-dimensional array. HFORMS(I) (integer) where I = 1,...,N aerial section numbers.
INITIALIZATION:	None
USERS:	SECPRM, OFFSET
LENGTH:	N

COMMON/HITP/

CONTENTS:	HITP contains sets of data used in computing hit probabilities of rounds following sensed misses and rounds following sensed hits.
ARRANGEMENT:	HITP is a two-dimensional array. HITP(I,J) (real, mils)

I = 1	subscript giving the total horizontal dispersion for a hit given a hit case for a stationary firer and a nonpinpoint situation
I = 2	subscript giving the total vertical dispersions for a hit given a hit case for a stationary firer and a nonpinpoint situation
I = 3	subscript giving the total horizontal dispersion for a hit given a sensed miss for a stationary firer and a nonpinpoint situation
I = 4	subscript giving the total vertical dispersion for a hit given a sensed miss for a stationary firer and a nonpinpoint situation
J = 1,...,N	where N is the number of firer-ammunition combination as defined by ICHAR in COMMON/IAMMO/.

INITIALIZATION: Required.

USERS: HPROB

LENGTH: 4*N

COMMON/HITPRB/

CONTENTS: HITPRB contains sets of horizontal and vertical fixes, biases and dispersions used in computing first round case hit probability for nonpinpoint situations.

ARRANGEMENT: HITPRB is a four-dimensional array.

HITPRB(I,J,K,L) (real, mils)

I = 1,...,6	subscripts giving the total horizontal dispersions corresponding to range intervals in COMMON/HPRNG/
I = 7,...,12	subscripts giving the total vertical dispersion corresponding to range intervals in COMMON/HPRNG/
I = 13,...,18	subscripts giving the total horizontal fixed biases corresponding to range intervals in COMMON/HPRNG/
I = 19,...,24	subscripts giving the total vertical fixed biases corresponding to range intervals in COMMON/HPRNG/
J = 1,...,6	subscripts corresponding to target speeds as defined in COMMON/TARSPD/
K = 1,...,7	subscripts corresponding to firer-target aspect as defined in COMMON/TARASP/
L = 1,...,M	subscripts corresponding to the firer-ammunition combination (e.g., ICHAR as defined in COMMON/IAMMO/) where M is the maximum value of ICHAR.

INITIALIZATION: Required.

USERS: DDSX3

LENGTH: 24*6*7*M

COMMON/HSPEED/

CONTENTS: HSPEED contains the desired speed of an aerial unit.

ARRANGEMENT: HSPEED is a one-dimensional array.

HSPEED(I,J) (real, meters/second)

where

$$I = \begin{cases} 1 & \text{unit is performing enroute movement} \\ 2 & \text{unit is searching for a target} \\ 3 & \text{unit is loitering} \end{cases}$$

J = 1,...,MWAIR-MWMIS aerial unit weapon codes (see COMMON/CBERR/ for definition of weapon code).

INITIALIZATION: Required

USERS: HFORM

LENGTH: 3 * (MWAIR-MWMIS)

COMMON/HXRT/

CONTENTS: HXRT contains the X coordinates of the movement control points for the lead element of each aerial section.

ARRANGEMENT: HXRT is a two-dimensional array.

HXRT(I,J) (real, meters)

where

I = 1,...,20 indices for the points in the route (see COMMON/XRT/ for arrangement)

J = 1,...,N aerial section numbers where N is the number of aerial sections.

INITIALIZATION: None

USERS: HXYMCP, RTSECT, RTLOIT

LENGTH: 20 * N

COMMON/HYRT/

CONTENTS: HYRT contains the Y analog of HXRT.

ARRANGEMENT: See COMMON/HXRT/.

INITIALIZATION: None

USERS: HXYMCP, RTSECT, RTLOIT

LENGTH: 20 * N

COMMON/HZRT/

CONTENTS: HZRT contains the Z analog of HXRT and is measured relative to the zero elevation plane.

ARRANGEMENT: See COMMON/HXRT/.

INITIALIZATION: None

USERS: HXYMCP, RTSECT, RTLOIT

LENGTH: 20 * N

COMMON/IANYPT/

CONTENTS: IANYPT contains data items used in analysis of the output of the simulation. This common area is filled during execution by a standard FORTRAN reading of data.

ARRANGEMENT: IANYPT contains four items as follows:

1. IRAND - the initial random number seed to be used (see COMMON/IRANDX/) (integer)
2. IPOINT - the initial FORTRAN unit on which analysis data is to be written (usually unit 1). If IPOINT is equal to zero, no analysis data will be written (integer)
3. IPTS(I) for I = 1,...,4 contains analysis points in the form of percentage of blue casualties. When a casualty point is reached, IPOINT is incremented by one to write the next set of analysis data on the next numbered FORTRAN unit
4. IPKMIN - the minimum kill type to be analyzed.

INITIALIZATION: Zero, altered during execution.

USERS: MAIN, FIRMOD

LENGTH: 7

COMMON/IBTAMO/

CONTENTS: IBTAMO contains the ammunition code for an artillery firing battery for assessing lethality of a counterbattery fire.

ARRANGEMENT: IBTAMO is a two-dimensional array. (integer*2)

IBTAMO(I, J) contains the ammo code for assessing lethality to the I^{th} battery by the J^{th} artillery weapon code for I = 1,..., 8; J = 1,..., MWART where MWART is the maximum artillery weapon code.

INITIALIZATION: Required, not altered during execution

USERS: ARLETH, FLIGHT

LENGTH: (8*MWART), H

COMMON/ICAP/

CONTENTS:	ICAP contains data used in the movement controller.
ARRANGEMENT:	ICAP has two values as follows:

1. The number of points in the maneuver unit or aerial section axis of advance (integer).

2. The number of points in the optimal route of advance (integer).

INITIALIZATION:	None	
USERS:	HELCON	RTLOIT
	MVCON	RTSEL
	PICKRT	RTSELH
	RTCROS	RTSRCH
LENGTH:	2	

COMMON/ICAP2/

CONTENTS: ICAP2 gives the number of points in an aerial section's route. This is the number of valid points in HXRT, HYRT, HZRT.

ARRANGEMENT: ICAP2 is a one-dimensional array.

ICAP2(I) (integer)

where

I = 1,...,N aerial section numbers and N is the total number of aerial sections.

INITIALIZATION: None

USERS: HXYMCP, RTSECT

LENGTH: N

COMMON/ICECOM/

CONTENTS: ICECOM contains data pertaining to the current element. It is set up by GETICE at the beginning of every element.

ARRANGEMENT: All data apply to the current element. ICECOM contains 32 values as follows:

1.	ICE	current element number (integer)
2.	XE	x-coordinate (real, meters)
3.	YE	y-coordinate (real, meters)
4.	SPD	speed (real, meters/second)
5.	DIR	direction of motion (real, radians)
6.	NSECT	section number (integer)
7.	NPOS	position in section (integer)
8.	KILL	kill code (integer)
9.	KOLOR	side (0 = blue, 1 = red, integer)
10.	IFPC	fire position code (0 = not in fire position, 1 = in fire position, integer)
11.	IDPC	desired position code (0 = has not reached desired position, 1 = has reached desired position, integer)
12.	MOBT	mobility type code (integer)
13.	NPLAT	platoon number (integer)
14.	EMICRT	micro-terrain elevation (real, meters)
15.	LSTGT	target element number (integer)
16.	IFIR	firing flag (0 = not firing, 1 = direct fire, 2 = indirect fire, integer)
17.	IFEELE	element number of first enemy element (integer)
18.	LEELE	element number of last enemy element (integer)
19.	MANUN	maneuver unit number (integer)
20.	NEUT	not used
21.	KWCOD	weapon type code (integer)
22.	TIME	length of last event (real, seconds)
23.	LPNPT	element number of the pinpointed element
24.	KFUNC	function code (1 = launcher, 2 = forward observer, 0 = neither, integer)
25.	NUM	launcher number or forward observer number depending on KFUNC (integer)
26.	CLOCKT	clock time (real, seconds)
27.	LWPNPT	element number of waiting pinpoint (integer)
28.	LFELTK	firing event list target element number (integer)
29.	EFELCK	firing event list clock time (real, seconds)

COMMON/ICECOM/ con't.

30.	KOMP	platoon net flag (1 = on platoon net, 0 = not on platoon net, integer)
31.	NAT	aerial platform maneuver unit number if the maneuver unit of ICE consists of aerial platforms, 0 otherwise (integer)
32.	NASEC	aerial section number if ICE is in an aerial section, 0 otherwise (integer)

INITIALIZATION: None

USERS:

ADTGSL	DELETH	HRAPUP	RETIRE
AFDC	DELETK	HXYMCP	REVAL
AFO	DETH	INTELL	RTATAK
AFSC	DETSQ	ISTHFF	RTCROS
AIRFB	DETT	ITRATE	RTJOIN
AIRFIR	DRFROK	LAUNCH	RTLOIT
AIRPOR	ELEPOS	LOIMOV	RTSECT
APFDYS	FIRCON	LOOK	RTSEL
APRIOR	FIRFOR	LPRENV	RTSELH
ARFO	FIRMOD	MICONC	RTSRCH
ATAN5	FLGSET	MOUNT	SECPRM
ATKPRM	FLIGHT	MOV	SENDI
AVAIL	FRMLDR	MVCON	SETLOS
CASELE	FSCHEK	MVTEST	SETRT
CBOBS	FTIME	MXMNRT	SETUP
CNTOUT	GETHEL	NATLDR	TACCNT
CON	GETICE	NEUDET	TDES
CONVRT	HDIF	NEWFO	TFCOMP
COV	HELCON	NEWMIS	TIME
CSMOVE	HELFIR	NUTARG	WPRIOR
CSRTSL	HELMOV	OFFSET	WRTFRL
CSWCAS	HFIRE	PICKRT	WRTIJ
CSWCON	HLNCH	PILOT	XYLOC
DEFPOS	HPROB	PREILL	XYLOCH
			XYMCP

LENGTH: 32

COMMON/ICSWEL/

CONTENTS: ICSWCL contains the element numbers of a crew served weapon group's carrier and crews. This common is used only when "analysis point" processing is desired (see COMMON/IANYPT/).

ARRANGEMENT: ICSWEL is a two-dimensional array

ICSWEL (I, J)

I = 1 contains the element number of the carrier in group J

I = 2, 3, 4 contains the element numbers of crews in group J. A value of 0 indicates no crew in this position of the group.

J = 1, ... N where N is the number of CSW groups.

INITIALIZATION: Required if "analysis point" processing is desired.

USERS: FIRMOD

LENGTH: 4*N

COMMON/ICSWGP/

CONTENTS: ICSWGP contains the crew served weapon group number of each element. This common is used only when "analysis point" processing is desired (see COMMON/IANYPT/).

ARRANGEMENT: ICSWGP is a one-dimensional array

ICSWGP (ICE) = CSW group number if ICE is an element of a CSW group, 0 otherwise

where ICE = 1, ..., MNELE, the maximum number of elements

INITIALIZATION: Required, if "analysis point" processing is desired

USERS: FIRMOD

LENGTH: MNELE

COMMON/IFBMIS/

CONTENTS: IFBMIS stores an array containing the mission flag of the firing team (integer).

ARRANGEMENT:

$$IFBMIS(I) = \begin{cases} 0 & \text{if firer I has no activity} \\ 1 & \text{if firer I has a confirmed target} \\ 2 & \text{if firer I has an unconfirmed target} \\ 3 & \text{if firer I is standing by for confirmation} \end{cases}$$

where I = 1,...,NTFSFT fire support firing team number.

INITIALIZATION: Zero, altered during execution.

USERS: AFO, CNFLCT, DRFROK, AIRFB, PRMSET, HELFIR, MFB

LENGTH: NTFSFT

COMMON/IFDCNT/

CONTENTS: IFDCNT stores an array containing the fire direction center radio network flag (integer).

ARRANGEMENT:

$$IFDCNT(I) = \begin{cases} 0 & \text{radio net I is open,} \\ 1 & \text{initial fire request is being sent on radio net I,} \\ 2 & \text{fire adjustment message is being sent on radio net I, and} \\ 3 & \text{fire request was communicated while receiver facilities were overloaded} \end{cases}$$

where I = 1,...,NTFRNT fire request net number.

INITIALIZATION: Zero, altered during execution.

USERS: AFO, FSCMON, AFDC, AFSC, AIRFB.

LENGTH: NTFRNT

COMMON/IFMC/

CONTENTS: IFMC contains the ammunition code of the MISTIC missile loaded aboard a launcher.

ARRANGEMENT: IFMC is a one-dimensional array.

IFMC(I) (integer)

where

I = 1,...,MWMIS-MWART; MISTIC unit weapon codes

INITIALIZATION: Required

USERS: MFB, LOADM

LENGTH: MWMIS - MWART

COMMON/IFRFL/

CONTENTS: IFRFL stores an array containing the activity flag of the forward observer or launcher vehicle (integer).

ARRANGEMENT:

IFRFL(I) = {
- 0 if all fire support activities are allowed for FO or launcher,
- 1 if all fire support activities are suspended for FO or launcher,
- 2 if launcher may load but not launch missiles, and
- 3 if launcher may launch but not load missiles

where

I = 1,...,ITOTFO for FO's
I = ITOTFO+1,...,ITOTFO+ITOTLN for launchers.

INITIALIZATION: Zero, changed during execution.

USERS: AFO, AFSC, DARFO, MFB

LENGTH: ITOTFO + ITOTLN

COMMON/IHAMO/

CONTENTS: IHAMO contains an array specifying the ammunition codes to be used by a helicopter during each phase of an attack.

ARRANGEMENT: IHAMO is a two-dimensional array.

IHAMO(I, J) (integer*2)

where

I =
- 1 suppressive fire weapon one, during attack phase 1,
- 2 suppressive weapon one, during attack phase two,
- 3 suppressive weapon two, during attack phase one,
- 4 suppressive weapon two, during attack phase two, and
- 5 point fire weapon, at end of attack phase one.

J = 1, . . . , MAXNAV helicopter element numbers.

INITIALIZATION: None

USERS: ATKPRM, CONVRT, HFIRE, RTATAK, TFCOMP

LENGTH: *(5*MAXNAV), H*

COMMON/IHDFMC/

CONTENTS: IHDFMC contains the ammunition code of the direct-fire missile loaded on a helicopter.

ARRANGEMENT: IHDFMC is a two-dimensional array.

IHDFMC(I, J) (integer)

where

I
- 1 MISTIC missile
- 2 beam rider missile

J = 1,..., NA3, where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,..., MAXLWC for ground elements and from MAXLWC + 1,..., MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: APRIOR, ATKPRM, HFIRE, HLNCH, RTATAK, TFCOMP

LENGTH: 2*NA3

COMMON/IHFNA/

CONTENTS: IHFNA contains an array specifying the number of the formation pattern to be used by an aerial unit under different circumstances.

ARRANGEMENT: IHFNA(I,J,K) (integer)

where

$$I = \begin{cases} 1 & \text{enroute movement,} \\ 2 & \text{loiter, and} \\ 3 & \text{search.} \end{cases}$$

$$J = \begin{cases} 1 & \text{for section one in a platoon,} \\ 2 & \text{for section two in a platoon,} \\ 3 & \text{for platoons in a team, and} \\ 4 & \text{for teams.} \end{cases}$$

K = 1,2,...,MWAIR-MWMIS aerial unit weapon codes.

INITIALIZATION: Required

USERS: HFORM

LENGTH: 3 * 4 * (MWAIR-MWMIS)

COMMON/IHTARG/

CONTENTS: IHTARG contains an array specifying the target elements to be attacked by a helicopter during each phase of an attack.

ARRANGEMENT: IHTARG is a two-dimensional array.

IHTARG(I, J) = element number to be attacked (integer*2)

where

I =
1. suppressive fire weapon one, during attack phase one,
2. suppressive fire weapon one, during attack phase two,
3. suppressive fire weapon two, during attack phase one,
4. suppressive fire weapon two, during attack phase two, and
5. point fire weapon at end of attack phase one.

J = 1, . . . , MAXNAV helicopter element numbers

INITIALIZATION: None

USERS: ATKPRM, CONVRT, HFIRE, RTATAK, TFCOMP

LENGTH: (5*MAXNAV), H

COMMON/INPRIR/

CONTENTS: INPRIR contains the priority of indirect-fire support missions assigned to aerial teams.

ARRANGEMENT: INPRIR is a two-dimensional array.

INPRIR(I, J) (integer)

where

$$I = \begin{cases} 1 \text{ for MISTIC launcher mission,} \\ 2 \text{ for MISTIC FO mission, and} \\ 3 \text{ for special FO mission.} \end{cases}$$

$$J = \begin{cases} 1 \text{ for blue force, and} \\ 2 \text{ for red force.} \end{cases}$$

INITIALIZATION: Required

USERS: AFSC

LENGTH: 3 * 2

COMMON/INTRIG/

CONTENTS: Six cells containing the parameters for the trigger area table for the fire support target selection model in integer mode.

ARRANGEMENT:

NRFTRG contains the number of red trigger areas triggered by red elements (≤ NRTRIG)

NRTRIG contains the total number of red trigger areas (≤ 10)

NBFTRG contains the number of blue trigger areas triggered by blue elements (≤ NBTRIG)

NBTRIG contains the total number of blue trigger areas (≤ 10)

NBATRG contains the number of search and destroy trigger areas triggered by blue elements (≤ NATRIG)

NATRIG contains the total number of search and destroy trigger areas triggered by friendly elements (≤ 10).

INITIALIZATION: Input, not altered during execution.

USERS: AFO, FSCHEK, TRIG

LENGTH: 6

COMMON/IPHASE/

CONTENTS: IPHASE stores an array containing the mission flag of each aerial team (integer).

ARRANGEMENT:

$$\text{IPHASE(I)} = \begin{cases} 0 & \text{aerial team I is in a mission operations area,} \\ 1 & \text{aerial team I is in transit to a mission operations area, and} \\ 2 & \text{aerial team I has just arrived in the vicinity of the battlefield} \end{cases}$$

where I = 1,...,NUMAVT aerial team number.

INITIALIZATION: A value of two is required.

USERS: AVAIL, ATDEC, AIRFB, NEWMIS, FLGSET, HELMOV, PRMSET, SECSET, FRMLDR, HXYMCP, HELFIR, DARFO.

LENGTH: NUMAVT

COMMON/IREQ/

CONTENTS: IREQ stores the number of the request on the ground-to-air communicator's fire request list which the communicator desires to transmit.

ARRANGEMENT: IREQ(I) (integer)

where

$$\text{I} = \begin{cases} 1 & \text{blue force communicator, and} \\ 2 & \text{red force communicator.} \end{cases}$$

INITIALIZATION: Zero, altered during execution.

USERS: AFSC

LENGTH: 2

COMMON/IRFNH/

CONTENTS: IRFNH contains a flag indicating if an aim adjustment event for a rapid fire weapon did not result in a hit within the allotted event time.

ARRANGEMENT:

$$IRFNH(I) = \begin{cases} 0 & \text{if aim adjustment resulted in a hit, and} \\ 1 & \text{if no hit resulted in aim adjustment (integer)} \end{cases}$$

for I = 1,...,N where N is the number of regular elements.

INITIALIZATION: None

USERS: FIRCON, FTIME

LENGTH: N

COMMON/ISACT/

CONTENTS: ISACT contains the section activity code indicating that a reevaluation of the section's firing position is required. A 1 if the firing position reevaluation is required; a 0, otherwise. For aerial vehicle sections, the section activity code indicates that a movement decision for the section should be made.

ARRANGEMENT: ISACT is a one-dimensional array.

ISACT(I) (integer)

I = 1,...,N section numbers

where N is the total number of sections.

INITIALIZATION: None

USERS: FIRKIL, CASELE, FIRFOR, FIRCON, DELETK, MFB, LMUFL, HELCON

LENGTH: N

COMMON/ISCHPR/

CONTENTS: ISCHPR contains the priorities for artillery firing battery scheduled fires.

ARRANGEMENT: ISCHPR is a two-dimensional array.

ISCHPR(I,J) contains the priority for the I^{th} scheduled fire by the J^{th} battery for I = 1,...,6; J = 1,...,N where N is the number of batteries.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: 6 * N

COMMON/ISFNA/

CONTENTS: ISFNA contains the formation pattern numbers to be used by aerial section while performing independent movement.

ARRANGEMENT: ISFNA is a two-dimensional array.

ISFNA(I,J) (integer)

where

$$I = \begin{cases} 1 \text{ for enroute movement,} \\ 2 \text{ for loiter,} \\ 3 \text{ for target attack, and} \\ 4 \text{ for target search} \end{cases}$$

J = 1,...,N helicopter element weapon codes (see COMMON/ANGLIM/ for weapon code definition).

INITIALIZATION: Required

USERS: SECPRM

LENGTH: 4 * N

COMMON/ITMASS/

CONTENTS: ITMASS stores an array containing the number of the aerial team assigned to deliver fire for request J on the ground-to-air list I.

ARRANGEMENT: ITMASS is a two-dimensional array:

ITMASS(I, J) (integer)

where

I = 1	blue force
I = 2	red force
J = 1,...,5	position in fire request list

INITIALIZATION: Zero, altered during execution

USERS: AFASGN, AFSC, AIRFB, FSCMON, WRTFRL

LENGTH: 10

COMMON/ITRIG/

CONTENTS: ITRIG stores an array containing the priority weight of on-call fire triggered by friendly elements in trigger area I.

ARRANGEMENT: ITRIG(I) (integer)

where

I = 1,...,30 trigger area number

INITIALIZATION: Required

USERS: AFO, FSCHEK

LENGTH: 30

COMMON/ITYPA/

CONTENTS: ITYPA contains the ground weapon classification; i.e., a heavy or light weapon, for aerial vehicle target selection.

ARRANGEMENT: ITYPA is a one-dimensional array (integer*2)

$$\text{ITYPA}(I) = \begin{cases} 0 & \text{the } I^{th} \text{ element is a light weapon or an aerial platform, and} \\ 1 & \text{the } I^{th} \text{ element is a heavy weapon} \end{cases}$$

for I = 1,..., MNELE where MNELE is the total number of elements

INITIALIZATION: Required

USERS: AIRPOR, WASAIR, AIRFIR

LENGTH: MNELE, H

COMMON/IUNACT/

CONTENT: IUNACT specifies the mission an aerial unit is performing.

ARRANGEMENT: IUNACT is a one-dimensional array.

IUNACT(I) (integer)

0 no mission
1 target of opportunity
2 counterbattery attack
3 search and destroy
4 self defense
5 indirect-fire MISTIC launcher
6 MISTIC forward observer
7 counterbattery observation
8 special forward observer
9 defensive operation
10 loitering off battlefield or retiring

where I = 1,..., NUMAVT aerial unit numbers.

INITIALIZATION: Required

USERS: Aerial Vehicle Model, Fire Support Target Selection Model.

LENGTH: NUMAVT

COMMON/JAMOAV/

CONTENTS: JAMOAV contains the minimum ammunition requirement for a given aerial vehicle weapon code to conduct a fire mission.

ARRANGEMENT: JAMOAV is a two-dimensional array (integer*2)

JAMOAV(I, J) contains the amount of ammunition type I for the J^{th} aerial weapon code

for I = 1,..., 6 and
J = 1,..., NA3 where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,..., MAXLWC for ground elements and from MAXLWC + 1,..., MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: AIRAMO

LENGTH: (6*NA3), H

COMMON/JPHASE/

CONTENTS: JPHASE specifies the mission phase of an aerial section.

ARRANGEMENT: JPHASE is a one-dimensional array.

JPHASE(I) (integer)

1 if aerial section I is enroute
0 if aerial section I has achieved its objective

where I = 1,...,N aerial section numbers.

INITIALIZATION: None

USERS: Aerial Vehicle Model

LENGTH: N

COMMON/JUNACT/

CONTENTS: JUNACT specifies the present activity of an aerial section.

ARRANGEMENT: JUNACT is a one-dimensional array.

JUNACT(I) (integer)

0 if aerial section I is flying enroute with unit
1 if aerial section I is searching for a target or loitering with unit
2 if aerial section I is attacking a target
3 if aerial section I is seeking a holding position
4 if aerial section I is retiring from the battlefield independently
5 if the aerial section I is independently seeking a defensive position
6 if aerial section I is enroute to rejoin unit

where I = 1,...,N aerial section numbers.

INITIALIZATION: None

USERS: Aerial Vehicle Model

LENGTH: N

COMMON/KADV/

CONTENTS: KADV contains the phase line numbers triggering the advance of supporting-fire units.

ARRANGEMENT: KADV is a one-dimensional array

where

KADV(I) = the phase line number triggering the Ith advance of supporting-fire units (integer)

for I = 1,...,N where N is the total number of advances that supporting-fire units are to execute.

INITIALIZATION: Required if phase lines are incorporated.

USERS: MVCON

LENGTH: N

COMMON/KAMAVL/

CONTENTS: KAMAVL contains ammunition codes of each ammunition assigned to an aerial vehicle during a fire mission.

ARRANGEMENT: KAMAVL is a two-dimensional array. (integer*2)

KAMAVL contains the ammunition code assigned to the I^{th} weapon of the J^{th} aerial vehicle

where

$$I = \begin{cases} 1 & \text{point weapon (destructive)} \\ 2 & \text{first suppressive fire weapon} \\ 3 & \text{second suppressive fire weapon.} \end{cases}$$

J = 1,..., MAXNAV where MAXNAV is the maximum number of aerial vehicle elements.

INITIALIZATION: None

USERS: CONVRT, WASAIR

LENGTH: (3*MAXNAV), II

COMMON/KAMMAX/

CONTENTS: KAMMAX contains the weapon firing configuration flag for an aerial vehicle performing a firing mission.

ARRANGEMENT: KAMMAX is a one-dimensional array.

$$KAMMAX(I) \begin{cases} 0 & \text{vehicle can fire one point fire weapon only} \\ 1 & \text{vehicle can fire one point fire and one suppressive fire weapon} \\ 2 & \text{vehicle can fire one point fire and two suppressive fire weapons} \\ 3 & \text{vehicle can fire no point fire and one suppressive fire weapon} \\ 4 & \text{vehicle can fire no point fire and two suppressive fire weapons} \end{cases}$$

for I = 1,..., NA3 where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,..., MAXLWC for ground elements and from

MAXLWC + 1, . . . , MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC − MAXLWC.

INITIALIZATION: Required

USERS: WASAIR, AIRAMO

LENGTH: NA3

COMMON/KAMOAV/

CONTENTS: KAMOAV contains an ammunition category for an aerial vehicle weapon code and ammunition code.

ARRANGEMENT: KAMOAV is a two-dimensional array (integer*2)

$$KAMOAV(I,J) = \begin{cases} 0 & \text{suppressive area fire ammunition, and} \\ 1 & \text{point (destructive) fire ammunition} \end{cases}$$

for I = 1, . . . , 6 ammunition code, and
J = 1, . . . , NA3 where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1, . . . , MAXLWC for ground elements and from MAXLWC + 1, . . . , MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC − MAXLWC.

INITIALIZATION: Required

USERS: AIRAMO, WASAIR

LENGTH: (6*NA3), II

COMMON/KAMPRD/

CONTENTS: KAMPRD contains the ammunition priority list for an aerial vehicle dependent upon aerial vehicle weapon code, ammunition code, and a light or heavy target (see COMMON/ITYPA/).

ARRANGEMENT: KAMPRD is a three-dimensional array (integer*2)

KAMPRD(I, J, K) contains the priority for the I^{th} type of weapon and the J^{th} ammunition code for the K^{th} aerial weapon code

for

$$I = \begin{cases} 1 & \text{heavy weapon} \\ 2 & \text{light weapon} \end{cases}$$

J = 1,...,NAMMOS where NAMMOS is the number of ammunition codes allowed per element, a user supplied storage dimension parameter

K = 1,..., NA3 where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON/LWCOD/ are numbered 1,..., MAXLWC for ground elements and from MAXLWC + 1,..., MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: WASAIR

LENGTH: (2*NAMMOS*NA3), H

COMMON/KANCEL/

CONTENTS: KANCEL contains the fire support element number to which a cancellation message is to be sent by an aerial vehicle team.

ARRANGEMENT: KANCEL is a one-dimensional array.

KANCEL(I) (integer)

where

I = 1, 2,...,NUMAVT aerial vehicle unit numbers.

INITIALIZATION: None

USERS: AIRFB

LENGTH: NUMAVT

COMMON/KELAGN/

CONTENTS: KELAGN contains the enemy element number assigned to each weapon of each aerial vehicle.

ARRANGEMENT: KELAGN is a two dimensional array. (integer * 2)

KELAGN(I, J) contains the target for the I^{th} weapon of the J^{th} aerial vehicle

for

$$I = \begin{cases} 1 & \text{point weapon} \\ 2 & \text{first suppressive fire weapon} \\ 3 & \text{second suppressive fire weapon} \end{cases}$$

J = 1, . . . , MAXNAV where MAXNAV is the maximum number of aerial vehicles.

INITIALIZATION: None

USERS: CONVRT, WASAIR

LENGTH: 3 * MAXNAV, H

COMMON/KFDC/

CONTENTS: KFDC contains a casualty flag for each fire direction center as a result of a counterbattery mission.

ARRANGEMENT: KFDC is a one-dimensional array.

KFDC(I) (integer)

where

$$\mathrm{KFDC(I)} = \begin{cases} 0 & \text{fire direction center I has not been neutralized} \\ 1 & \text{the center is short-term neutralized} \\ 2 & \text{the center is long-term neutralized} \end{cases}$$

for I = 1,...,N where N is the maximum number of artillery fire direction centers.

INITIALIZATION: None

USERS: BTLETH, AFO, AFDC

LENGTH: N

COMMON/KFO/

CONTENTS: KFO contains the element for which a fire support firer is delivering fire or for aerial units performing a counter-battery mission, the battery against which the mission is directed.

ARRANGEMENT: KFO is a one-dimensional array.

KFO(I) (integer)

where

I = 1,...,NTFSFT

INITIALIZATION: None

USERS: MFB, AIRFB, PRMSET, CBCONT

LENGTH: NTFSFT

COMMON/KFOD/

CONTENTS: KFOD stores an array containing the event flag of each forward observer and the ground-to-air communicator.

$$KFOD(I) = \begin{cases} 0 & \text{forward observer I is to perform target selection activities} \\ 1 & \text{communicate initial target selection} \\ 2 & \text{observe for fire adjustment} \\ 3 & \text{communicate fire adjustment} \\ 4 & \text{try to communicate over a new net (integer)} \end{cases}$$

where

$$I = \begin{cases} 1, \ldots, \text{ITOTFO, the total number of forward observers, a user-specified storage dimension parameter} \\ \text{ITOTFO} + 1 \quad \text{blue ground-to-air communicator} \\ \text{ITOTFO} + 2 \quad \text{red ground-to-air communicator.} \end{cases}$$

INITIALIZATION: None

USERS: AFB, AFO, AFSC, AIRFB, ARFO, CENTER, COUNT, COV, FIRCON, FIRFOR, FSCMON, HELFIR, ITRATE, MFB, NEWFO, WRITFO

LENGTH: ITOTFO + 2

COMMON/KILFIR/

CONTENTS: KILFIR contains information about the effects of air defense weapon lethality on the firepower of helicopter elements.

ARRANGEMENT: KILFIR is a two-dimensional array.

KILFIR(I,J) (integer)

where

I = 1,...,MAXK indicates the type of damage inflicted by an air defense weapon

J = 1,...,N indicates the weapon code of the damaged helicopter (see COMMON/ANGLIM/ for a definition of helicopter weapon code)

AD-A040 053 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM). VOLUME 2, SECTION--ETC(U)
DEC 71 G M CLARK, R J WILHELM DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-2(U)-SECT-1 NL

2 OF 5
AD
A040053

OF 5

0053

and

MAXK is the maximum number of air defense lethality codes ($1 \leq MAXK \leq 9$)

Each entry in KILFIR is an integer of up to MNAMO digits defined as follows:

$$KILFIR(I,J) = \{ \delta_K ; K = 1, \ldots, MNAMO \}$$

where

$$\delta_K = \begin{cases} 1 & \text{if ammunition of type K can no longer be fired as a result of the damage} \\ 0 & \text{if otherwise} \end{cases}$$

(See COMMON/NUMBER/ for definition of MNAMO)

INITIALIZATION: Required

USERS: CASHEL

LENGTH: MAXK * N

COMMON/KLCPE/

CONTENTS: KLCPE contains the location in a list of the next movement control point for CSW team NCR.

ARRANGEMENT: KLCPE is a one dimensional array.

KLCPE(NCR) (integer)

where

NCR = 1, . . . , NCSWMX CSW team

INITIALIZATION: Zero, altered during execution.

USERS: CSMOVE, CSWGRD, CSWCON, MOUNT

LENGTH: NCSWMX

COMMON/KMANU/

CONTENTS: KMANU contains the maneuver unit number corresponding to an aerial unit or a MISTIC unit.

ARRANGEMENT: KMANU is a one-dimensional array.

KMANU(I) (integer)

where

$$I = \begin{cases} 1,2,\ldots,\text{NUMAVT} & \text{aerial unit numbers,} \\ \text{NUMAVT}+1,\ldots,\text{NUMAVT}+\text{MISTUN} & \text{MISTIC unit numbers} \end{cases}$$

INITIALIZATION: Required

USERS: AIRFB, SECSET, MISEND, COUNT, LOIPOS, LOIMOV, DEFSET, DEFPOS

LENGTH: NUMAVT + MISTUN

COMMON/KPATRN/

CONTENTS: KPATRN stores an array containing the firing pattern to be used in firing on each concentration I, given on-call fires were triggered by friendly elements.

ARRANGEMENT: KPATRN(I) (integer)

where I = 1,...,20 concentration number

INITIALIZATION: Required

USERS: AFO

LENGTH: 20

COMMON/KPHAS/

CONTENTS: KPHAS contains a flag indicating the status of each attacking maneuver unit with respect to the phase line being approached.

ARRANGEMENT: KPHAS is a one-dimensional array

where

$$KPHAS(I) = \begin{cases} 0 & \text{if the } I^{th} \text{ maneuver unit has not reached the phase line zone,} \\ 1 & \text{if the } I^{th} \text{ maneuver unit is within the phase line zone, and} \\ 2 & \text{if the } I^{th} \text{ maneuver unit has crossed the phase line but may not proceed with the attack (integer)} \end{cases}$$

for I = 1,...,N, where N is the number of maneuver units.

INITIALIZATION: None

USERS: MVCON

LENGTH: N

COMMON/KTUBE/

CONTENTS: KTUBE contains a lethality flag for each weapon in an artillery firing battery which becomes a casualty as a result of a counterbattery mission.

ARRANGEMENT: KTUBE is a two-dimensional array.

KTUBE(I,J) (integer)

where

$$
KTUBE(I,J) = \begin{cases} 0 & \text{weapon I in battery J is not neutralized} \\ 1 & \text{the weapon is short-term neutralized} \\ 2 & \text{the weapon is long-term neutralized} \end{cases}
$$

for I = 1,...,6
J = 1,...,N where N is the maximum number of artillery firing batteries.

INITIALIZATION: None

USERS: AFB, BTLETH

LENGTH: 6 * N

COMMON/KSALT/

CONTENTS: KSALT contains the status flag for a CSW team with respect to its alternate firing position.

ARRANGEMENT: KSALT is a one dimensional array.

KSALT(NCR) (integer)

where

NCR = 1, . . . , NCSWMX CSW team

$$
KSALT(NCR) = \begin{cases} 2 & \text{NCR is at its alternate firing position} \\ 1 & \text{NCR is moving to its alternate firing position} \\ 0 & \text{otherwise} \end{cases}
$$

INITIALIZATION: Zero, altered during execution

USERS: CSMOVE, CSWCON, MOUNT

LENGTH: NCSWMX

COMMON/KSDECN/

CONTENTS: KSDECN contains a flag indicating a CSW team's deployment mode.

ARRANGEMENT: KSDECN is a one dimensional array.

KSDECN(NCR) (integer)

where

NCR = 1, . . . , NCSWMX CSW team

$$KSDECN(NCR) = \begin{cases} 2 & \text{NCR is in defensive mode} \\ 1 & \text{NCR is in attack mode} \\ 0 & \text{otherwise} \end{cases}$$

INITIALIZATION: Zero, altered during execution

USERS: CSWCON, CSWGRD

LENGTH: NCSWMX

COMMON/KSDSMT/

CONTENTS: KSDSMT contains the indicator for the deployment of each APC's CSW teams.

ARRANGEMENT: KSDSMT is a one dimensional array.

KSDSMT(I) (integer)

where

I = 1, . . . , NAPCMX APC number

$$KSDSMT(I) = \begin{cases} 1 & \text{APC to deploy crews in attack mode} \\ 0 & \text{APC to deploy crews in defense mode} \end{cases}$$

INITIALIZATION: Required

USERS: CSWCAS, CSWCON

LENGTH: NAPCMX

COMMON/KSMAKE/

CONTENTS: KSMAKE contains the status flag with respect to a CSW team and its primary firing position.

ARRANGEMENT: KSMAKE is a one dimensional array.

KSMAKE(NCR) (integer)

where

NCR = 1, . . . , NCSWMX CSW team

$$\text{KSMAKE(NCR)} = \begin{cases} 2 & \text{NCR is at its primary firing position} \\ 1 & \text{NCR is moving to its primary firing position} \\ 0 & \text{otherwise} \end{cases}$$

INITIALIZATION: Zero, altered during execution

USERS: CSMOVE, CSWCON, MOUNT

LENGTH: NCSWMX

COMMON/KSMVT/

CONTENTS: KSMVT contains the number of rounds a crew is to fire before alternating its fire position. If crew is not to alternate firing positions, this value is set to zero.

ARRANGEMENT: KSMVT is a one dimensional array.

KSMVT(NCR) (integer)

where

NCR = 1, . . . , NCSWMX CSW team

INITIALIZATION: Required

USERS: CSWCON

LENGTH: NCSWMX

COMMON/KSWHCH/

CONTENTS: KSWHCH contains either the APC number of each regular element that is a carrier or the crew-served weapon (CSW) team number for elements that are crews.

ARRANGEMENT: KSWHCH is a one dimensional array.

KSWHCH(I) (integer)

I = 1, . . . , NUMELE, regular element numbers

when

LCSWFN(I) = 1, KSWHCH(I) = APC number

when

LCSWFN(I) = 2, KSWHCH(I) = CSW number

when

LCSWFN(I) = 0, KSWHCH(I) = 0.

INITIALIZATION: Required

USERS: CSMOVE, CSWCAS, CSWCON, MOUNT

LENGTH: NUMELE

COMMON/KWAVAL/

CONTENTS: KWAVAL contains the current status of the artillery unit.

ARRANGEMENT: KWAVAL(I) (integer)

where

I = 1,...,MWART artillery unit weapon code

$$\text{KWAVAL(I)} = \begin{cases} 0 & \text{indicates at least one artillery unit with artillery weapon code I is not a casualty} \\ 1 & \text{otherwise} \end{cases}$$

INITIALIZATION: Zero, may be altered during execution.

USERS: AFO, AFSC, FSCMON, REASFO

LENGTH: MWART

COMMON/LAMMO/

CONTENTS: LAMMO contains the ammunition supply for each element.

ARRANGEMENT: LAMMO is a two-dimensional array:

LAMMO (I,J) (integer)

I = 1,...,NAMMOS ammunition code numbers
J = 1,...,MNELE element numbers

where NAMMOS is the number of ammunition codes allowed per element and MNELE is the total number of elements (user supplied storage dimension parameters)

INITIALIZATION: Required

USERS: ADGCON, AMMO, AMMOD, AMMODR, CASHEL, LAUNCH, LOADM, PASMIS

LENGTH: NAMMOS * MNELE

COMMON/LAPCCW/

CONTENTS: Common LAPCCW stores the crew numbers of the CSW teams assigned to an APC carrier.

ARRANGEMENT: LAPCCW is a two-dimensional array:

LAPCCW(I,J) (integer)

where

I = 1,..., 4 CSW team
J = 1,..., NAPCMX APC carrier

where NAPCMX is the maximum number of APC's.

INITIALIZATION: Required

USERS CSWCON, CSWGRD

LENGTH: 4 * NAPCMX

COMMON/LASTFB/

CONTENTS: LASTFB contains the previous event number of each firing battery

(for output appearance only)

ARRANGEMENT: LASTFB is a one-dimensional array:

LASTFB (I) (integer)
I = 1,...,NTFSFT

where

NTFSFT is the number of fire-support units

NTFSFT = NUMART + ITOTLN + NUMAVT

INITIALIZATION: None

USERS: AFB, MFB, WRTART

LENGTH: NTFSFT

COMMON/LCB/

CONTENTS: LCB contains the enemy battery number of an artillery counterbattery fire mission presently being fired.

ARRANGEMENT: LCB is a one-dimensional array.

LCB(I) (integer)

where

$$LCB(I) = \begin{cases} \text{the enemy battery number of a counter-battery fire for battery I} \\ 0 \quad \text{a counterbattery mission is not being fired} \end{cases}$$

for I = 1,...,N where N is the number of batteries.

INITIALIZATION: Zero

USERS: AFB

LENGTH: N

COMMON/LCREW/

CONTENTS: LCREW contains the crew-served weapon mount/dismount status flag.

ARRANGEMENT: LCREW is a one-dimensional array.

LCREW(I)

$I = 1, \cdots, N$

where N = number of sections, and

LCREW(I) =

0 crews in section I are mounted
1 crews in section I are to dismount
2 crews in section I are dismounted
3 crews in section I are to mount

INITIALIZATION: Required

USERS: DISMNT, MOUNT, CARIER, CSWCAS, LOOK

LENGTH: N

COMMON/LCSWFN/

CONTENTS: LCSWFN contains the CSW function code for the DYNCOM element in question.

ARRANGEMENT: LCSWFN is a one-dimensional array.

$$\text{LCSWFN(I)} = \begin{cases} 2 \text{ if element } I \text{ is a CSW crew} \\ 1 \text{ if element } I \text{ is a CSW carrier} \\ 0 \text{ otherwise} \end{cases}$$

$I = 1, \cdots, N$

where N = last element number

INITIALIZATION: Required

USERS: CSWCAS, FIRMOD, LOOK

LENGTH: N

COMMON/LCSWTP/

CONTENTS: LCSWTP contains the CSW weapon code number for each team aboard a specified carrier.

ARRANGEMENT: LCSWTP is a one-dimensional array:

LCSWTP (N C R) (integer)
NCR = 1,...,NCSWMX CSW teams

where

NCSWMX is the total number of CSW teams

INITIALIZATION: Required

USERS: CSWCON, MOUNT

LENGTH: NCSWMX

COMMON/LFLAG/

CONTENTS: LFLAG contains codes to indicate the status of missiles already launched or to be launched by aerial platforms.

ARRANGEMENT: LFLAG is a one-dimensional array of integers.

$$\text{LFLAG(I)} = \begin{cases} 0 & \text{no significance} \\ 2 & \text{the missile launched by launcher I} \\ & \text{is still flying} \end{cases}$$

where I = 1,...,N and N is the maximum number of launchers.

INITIALIZATION: None

USERS: HELFIR, ATKPRM, LAUNCH, MFB

LENGTH: N

COMMON/LKILL/

CONTENTS: LKILL contains the kill code of each element. A 0 inicates no kill; a 1 is mobility kill; a 2 is firepower kill; a 3 is mobility and fire-power kill; and a 4 is complete kill.

ARRANGEMENT: LKILL is a linear array.

LKILL(I) (integer)

I = 1, ···, N element numbers

where N is the total number of elements.

INITIALIZATION: None

USERS: ISORT, ORGOUT, WRIDLS, FIRKIL, CASELE, IEXP, GETICE, FIRCON, DETSQ, INTELL, RTSEL, TACCNT, FORM, MXMNRT, MVCON, SEND, SETLOS, MAIN

LENGTH: N

COMMON/LMOVF

CONTENTS: LMOVF contains a movement flag for each element. A 1 if the element moved during his latest event; a 0 if the element remained stationary during his latest event.

ARRANGEMENT: LMOVF is a linear array.

LMOVF(I) (integer)

I = 1, ···, N element numbers

where N is the total number of elements.

INITIALIZATION: None

USERS: MVTEST, FIRCON

LENGTH: N

COMMON/LMUFL/

CONTENTS: LMUFL specifies whether or not a maneuver unit has a pending movement decision.

ARRANGEMENT: LMUFL is a one-dimensional array.

LMUFL(I) (integer)

where

$$\text{LMUFL(I)} = \begin{cases} 1 & \text{maneuver unit I has a pending movement decision} \\ 0 & \text{otherwise} \end{cases}$$

and I = 1,...,MNMNU maneuver unit numbers

INITIALIZATION: None

USERS: AIRFB, HELCON, HELFIR, LMUSET, MFB

LENGTH: MNMNU

COMMON/LNSET/

CONTENTS: LNSET contains data associated with the launch operation.

ARRANGEMENT: LNSET contains 18 values:

1. TM = time of missile launch (real, seconds).
2. X = X-coordinate of launch point (real, meters).
3. Y = Y-coordinate of launch point (real, meters).
4. Z = Z-coordinate of launch point (real, meters).
5. ITARG = missile target element number (integer).
6. NCE = launcher element number (integer).
7. LFO = FO element number (integer).
8. NUM = launcher's launcher number (integer).

9. IFO = FO's FO number (integer).

10. T = missile flight time (real, seconds).

11. FINFLG = flag to indicate call to flight (integer)

$$\text{FINFLG} = \begin{cases} 0 & \text{indirect-fire mission} \\ 1 & \text{direct-fire mission} \\ 2 & \text{continuation of previous event.} \end{cases}$$

12. TIMP = projected impact time (real, seconds).

13. ANGMIS = the angle of the tangent to the missile flight path at time T.

14. ANGLCH = the launch angle of the ballistic missile.

15. RMEAN = missile roll at beginning of this time interval (real, radians)

16. B1 = the interpolation factor for the replication.

17. CROLL = missile roll at the end of this time interval (real, radians)

18. MINUM = the number of the missile that has been launched.

INITIALIZATION: None

USERS: LAUNCH, FLIGHT, FINAL, MICONG

LENGTH: 18

COMMON/LNUM/

CONTENTS: LNUM contains the launcher number or forward observer number for each element if it is a launcher or forward observer, respectively.

ARRANGEMENT: LNUM is a linear array.

LNUM(I) (integer)

I = 1, ···, N element numbers

where N is the total number of elements.

INITIALIZATION: Required

USERS: FIRCON, COV, FIRKIL, GETICE, FIRFOR, DELETR, SENDFR

LENGTH: N

COMMON/LTHTNK/

CONTENTS: LTHTNK contains the last subscript in common areas PKTNK and TPMKH of the lethality characteristics set for a given firer-ammunition-target combination. The values of LTHTNK are arranged such that all tank target combinations are the lowest set of numbers (e.g., 1 through MTNKLT from COMMON/MAXWEP/) and nontank target combinations are the highest set of numbers (e.g., MTNKLT + 1 through MLTHNK from COMMON/MAXWEP/).

ARRANGEMENT: LTHTNK is a three-dimensional array.

LTHTNK(I,J,K) (integer)

I = 1,...,6 - ammunition type of firer
J = 1,...,MNFRT - firer weapon codes
K = 1,...,MNFRT - target weapon codes

where MNFRT is the total number of weapon codes as defined in COMMON/IPRJCT/.

INITIALIZATION: Required

USERS: FIRCON, HFIRE

LENGTH: 6 * MNFRT * MNFRT

Note: To prevent COMMON/LTHTNK/ from being scenario dependent and to simplify preparation of input data, a currently used expedient fixes the upper bound on the second subscript at 25. Though some core is thus wasted, correct results will be obtained provided MNFRT ≤ 25. When used in this way the length declaration becomes 6 * 25 * MNFRT.

COMMON/LWCOD/

CONTENTS: LWCOD contains weapon code numbers for various DYNTACS X elements and fire support units.

ARRANGEMENT: LWCOD is a linear array of integers arranged as follows:

Element Class	I	Value Order Required
Vehicular	1 : N1	$1 \leq LWCOD(I) \leq MAXLWC$ for ground vehicles $LWCOD(I) > MAXLWC$ for aerial vehicles
Blue Artillery Units	N1 + 1 : N2	$0 \leq LWCOD(I) \leq MWBART$
Red Artillery Units	N2 + 1 : N3	$MWBART \leq LWCOD(I) \leq MWART$
Blue MISTIC Units	N3 + 1 : N4	$MWART \leq LWCOD(I) \leq MWBMIS$
Red MISTIC Units	N4 + 1 : N5	$MWBMIS \leq LWCOD(I) \leq MWMIS$
Blue Aerial Teams	N5 + 1 : N6	$MWMIS \leq LWCOD(I) \leq MWBAIR$
Red Aerial Teams	N6 + 1 : N7	$MWBAIR \leq LWCOD(I) \leq MWAIR$

where

N1 = NUMELE
N2 = N1 + NBART
N3 = N1 + NUMART
N4 = N3 + NBMIS
N5 = N3 + MISTUN
N6 = N5 + NBAVT
N7 = N5 + NUMAVT

and

MAXLWC, MWBART, MWART, MWBMIS, MWMIS, MWBAIR, MWAIR, NUMELE, NBART, NUMART, NBMIS, MISTUN, NBAVT and NUMAVT are as defined in common area /NUMBER/.

INITIALIZATION: Required

USERS: Widely used throughout the simulation.

LENGTH: NUMELE + NUMART + MISTUN + NUMAVT

COMMON/LWSYS/

CONTENTS: LWSYS contains the weapon system code for each weapon code.

ARRANGEMENT: LWSYS is a one-dimensional array:

LWSYS (integer)

$$
LWSYS(I) = \begin{cases} 1 & \text{if tank} \\ 2 & \text{if amored personnel carrier} \\ 3 & \text{if crew served weapon} \\ 4 & \text{if light armor} \\ 5 & \text{if AH56 helicopter} \\ 6 & \text{if light observation helicopter} \end{cases}
$$

I = 1, . . . , N weapon code numbers
where N is the total number of weapon codes.

INITIALIZATION: Required for each weapon code.

USERS: ADTGSL, AFO, APRJOR, ARLETH, FIRCON, FIRMOD, FLIGHT, HFIRE , REVAL, SELECA

LENGTH: N

COMMON/LZZPE/

CONTENTS: LZZPE contains a flag indicating that an element is ahead of the maneuver unit leader due to a kill to a previous leader.

ARRANGEMENT:

LZZPE(I) = 0 element I is not interpolating ahead of the new leader

LZZPE(I) ≠ 0 otherwise

for I = 1,...,N where N is the number of regular elements.

INITIALIZATION: None

USERS: XYMCP, INTERP

LENGTH: N

COMMON/MAINPR/

CONTENTS: MAINPR contains constants used to control overall program execution and the summary output interval.

ARRANGEMENT:

MAXEV	Maximum number of events to be executed. Execution is terminated when this limit is exceeded (integer)
OTIME	Time interval (battle time not real time) for summary outputs. Must be greater than EVTIME (see MOVPAR) (real, seconds)
KNTR	Current event counter (initialize to zero).

INITIALIZATION: Required

USERS: Main program, GETPUT

LENGTH: 3

COMMON/MANACT/

CONTENTS: MANACT contains flags indicating whether or not routes are to be chosen for aerial units.

ARRANGEMENT: MANACT is a one-dimensional array.

MANACT(I) (integer)

0 if a route is not to be chosen for unit I
IUNACT(I) + 1 if a route is to be chosen for unit I to perform the activity specified by IUNACT(I)

where I = 1,...,NUMAVT aerial unit numbers (See COMMON/IUNACT/).

INITIALIZATION: None

USERS: AIRFB, NEWMIS, HELCON

LENGTH: NUMAVT

COMMON/MAXWEP/

CONTENTS: MAXWEP contains dimensioning constants for common areas FIRKON, AMMOCH, and TPMKH, and constraint constants for common areas IHTPRB and LTHTNK.

ARRANGEMENT: MAXWEP contains seventeen values as follows:

1. MDIRFI = the number of direct-fire (e.g., main gun) projectile types being played in the battle simulation (integer).

2. MRAPFI = the number of rapid-fire projectile types being played in the battle simulation (integer).

3. MMIS = the number of beam-rider missile (e.g., Shillelagh) types being played in the battle simulation (integer, currently no greater than two).

4. MLTHNK = largest value of LTHTNK(I,J,K) (from COMMON/LTHTNK/) entered for rapid fire projectiles (integer).

5. MLTRAP = largest value of LTHTNK(I,J,K) (from COMMON/LTHTNK/) entered for rapid fire projectiles (integer).

6. MLTMIS = largest value of LTHTNK(I,J,K) (from COMMON/LTHTNK/) entered for direct-fire missiles. This value also specifies the total length of LTHTNK (integer).

7. MTNKLT = largest value of LTHTNK(I,J,K) against tanks.

8. MCGIW = number of ground MISTIC weapons.

9. MCGPW = number of air-defense missile weapons.

10. MPHM = number of air-defense gun systems.

11. MSAHM = number of aerial vehicle weapons.

12. MIPH = smallest value of firer weapon-ammunition code for passive homing missiles (see COMMON/IHTPRB/).

13. MXPH = largest value of firer weapon-ammunition code for passive homing missiles.

14. MISAH = smallest value of firer weapon-ammunition code for semi-active homing missiles.

15. MXSAH = largest value of firer weapon-ammunition code for semi-active homing missiles.

16. MICGI = smallest value of firer weapon-ammunition code for conventional guns with impact fused warheads.

17. MXCGI = largest value of firer weapon-ammunition code for conventional guns with impact fused warheads.

INITIALIZATION: Required

USERS: MAIN PROGRAM, ADGCON, ADTIC, AMMOD, AMMODR, FIRMOD, FTIME, GLOBE, HHPROB, HPROB, TLETH

LENGTH: 17

COMMON/MCLASS/

CONTENTS: MCLASS contains the class of mission that aerial units are currently performing.

ARRANGEMENT: MCLASS is a one-dimensional array.

MCLASS(I) (integer)

where

$$\text{MCLASS(I)} = \begin{cases} 1 & \text{aerial unit I is performing no mission} \\ 2 & \text{aerial unit I is performing an attack mission} \\ 3 & \text{aerial unit I is performing an observation mission} \\ 4 & \text{aerial unit I is performing an indirect-fire MISTIC mission} \end{cases}$$

where I = 1,...,NUMAVT aerial unit numbers.

INITIALIZATION: Required (should agree with initial values assigned in COMMON/IUNACT/).

USERS: AIRFB,RTCROS,RTATAK,HELFIR,NEWMIS,HELCON, MISEND,ATKPRM

LENGTH: NUMAVT

COMMON/MD/

CONTENTS: MD contains the standard deviations in flight time at a distance RM (given in COMMON/RM/).

ARRANGEMENT: TD(I) (real, seconds^2)

I = 1, ···, N

where N = number of distance points and

TD(1) = 0.0

INITIALIZATION: Required

USERS: FLIGHT

LENGTH: N

COMMON/MDFCSW/

CONTENTS: Common MDFCSW contains a flag which indicates whether a specified APC may fire when its crews are dismounted.

ARRANGEMENT: MDFCSW is a one-dimensional array (integer)

$$MDFCSW(I) = \begin{cases} 1 & \text{indicates firing allowed} \\ 0 & \text{otherwise} \end{cases}$$

for I = 1,...,NAPCMX, where NAPCMX is the maximum number of APC's.

INITIALIZATION: Required

USERS: CSWCON

LENGTH: NAPCMX

COMMON/MESARR/

CONTENTS: MESARR contains the number of fire requests awaiting action by an aerial unit.

ARRANGEMENT: MESARR is a one-dimensional array.

MESARR(I) (integer)

where

I = 1,...,NUMAVT aerial unit numbers

INITIALIZATION: None

USERS: AIRFB

LENGTH: NUMAVT

COMMON/MESCON/

CONTENTS: MESCON contains the communication status of an aerial unit during new mission negotiations.

ARRANGEMENT: MESCON is a one-dimensional array.

MESCON(I) (integer)

$$\mathrm{MESCON(I)} = \begin{cases} 0 & \text{no message has been received} \\ 1 & \text{the mission request has been confirmed} \\ 2 & \text{the mission request has been cancelled} \end{cases}$$

where I = 1,2,...,NUMAVT aerial unit numbers.

INITIALIZATION: None

USERS: AIRFB

LENGTH: NUMAVT

COMMON/MFCSW/

CONTENTS: Common MFCSW contains a flag which indicates whether a specified APC may fire when its crews are mounted.

ARRANGEMENT: MFCSW is a one-dimensional array (integer)

$$\mathrm{MFCSW(I)} = \begin{cases} 1 & \text{indicates firing allowed} \\ 0 & \text{otherwise} \end{cases}$$

for I = 1,...,NAPCMX, where NAPCMXis the maximum number of APC's.

INITIALIZATION: Required

USERS: CSWCON

LENGTH: NAPCMX

COMMON/MIDATA/

CONTENTS: MIDATA contains assorted indirect-fire (MISTIC) missile flight data.

ARRANGEMENT: MIDATA consists of 42 values:

1. SDXFI = standard deviation of the FO location error in the X direction when he is in his initial position (meters)

2. SDYFI = same as above in Y direction

3. SDXFM = standard deviation of the FO location error in the X direction after he has moved from his initial position (meters)

4. SDYFM = same as above in Y direction

5. SDXLI = standard deviation of the launcher's location error in the X direction when he is in his initial position (meters)

6. SDYLI = same as above in Y direction

7. SDXLM = standard deviation of the launcher's location error in the X direction after he has moved from his initial position (meters)

8. SDYLM = same as above in Y direction

9. DISMAX not currently used

10. RE = probability that a missile will be mechanically reliable in flight

11. CRD = standard deviation of the angular cross range bias in mils

12. CTNA = cotangent of the seeker angle which determines the rear limit of the field of view

13. CTNB = cotangent of the seeker angle which determines the forward limit of the field of view

14. TANGI = tangent of the seeker angle which determines the fringe region of the field of view

15. TANG2 = tangent of the seeker angle which determines the center region of the field of view

16. EPT = if the missile and target's clock times are within EPT of being equal, they are considered to be equal

17. EPS = the maximum absolute deviation between the forward edge of the field of view and the target's position within which the target is considered to be at the forward edge of the field of view

18. NPTS = the number of points in COMMON/DDPTS/ and COMMON/DTPTS/

19. RHO = autocorrelation between ballistic missile roll errors separated by DELT (real)

20. ROLLRT = standard deviation of ballistic missile roll error (real, radians)

21. DELT = time between target search checks for ballistic missiles (seconds)

22. AYP = absolute value of target y coordinate (meters, initialize to 0.0)

23. TIMLIM = minimum limiting time remaining before impact for the FO to change targets (seconds)

24. TIMENP = time interval between illuminator pulses (seconds)

25. ALIM = the limiting FO-target-missile angle (radians)

26. BLIM = the limiting FO-target-missile angle for marginal FO safety (radians)

27. MV = current missile velocity in meters/second (real, initialize to 0.0)

28. NSRPTS = number of points in COMMON/SRPTS/

29. NDDPTS = number of points in COMMON/DDPTS/

30. TANETA = tangent of the angle used in determining the limiting range from the target along the flight path beyond which the missile will home on the FO

31. IPREVT = previous missile event if this event is a continuation; initialize as 0.0

32. NPOINT = number of points in COMMON/RM/

33. NL = maximum number of missed pulses

34. YER = lateral error between target position and field of view (initialize as 0.0)

35. TVFIX = time interval between visual fixations for detection by an E-O device (seconds)

36. FOY2 not currently used

37. FOX3 not currently used

38. FOY3 not currently used

39. EPSR if the slant range to the target is less than the distance the missile will travel during the next time increment plus EPSR, then the missile is considered to impact during the next time interval (meters)

40. LD = $\begin{cases} 0 & \text{for first flight iteration} \\ 1 & \text{for iteration after first} \\ 2 & \text{for iteration to place target on edge of field of view} \end{cases}$

initialize as 0.

41. KEY = dummy variable used to pass alphabetic data between subroutines FLIGHT and FINAL

42. BXLDIS = target lead distance (meters).

INITIALIZATION: Required

USERS: BFLITE, FINAL, FINALE, FLIGHT, HLNCH, LAUNCH, MICONG, PROB, PROBTV, SEEKER, TVDET

LENGTH: 42

COMMON/MIFO/

CONTENTS: MIFO contains data associated with the missile flight procedure.

ARRANGEMENT: MIFO contains 29 values

1. XT = target's X-coordinate (real, meters)

2. YT = target's Y-coordinate (real, meters)

3. ZT = target's Z-coordinate (real, meters)

4. ANGT = target's direction of travel (real, radians)

5. SINT = sine of ANGT (real)

6. COST = cosine of ANGT (real)

7. HTV = height of the target vehicle (real, meters)

8. PCT = fraction of target not visible to the FO (real)

9. NFO = indicator whether FO lost sight of the target on the current iteration (pulse, integer)

$$\text{NFO} = \begin{cases} 0 \sim \text{FO lost sight of target this pulse} \\ 1 \sim \text{otherwise} \end{cases}$$

10. XFO = FO's X-coordinate (real, meters)

11. YFO = FO s Y-coordinate (real, meters)

12. ZFO = FO's Z-coordinate (real, meters)

13. R = range from the FO to the target (real, meters)

14. XM = missile's X-coordinate (real, meters)

15. YM = missile's Y-coordinate (real, meters)

16. ZM = missile s Z-coordinate (real, meters)

17. LM = indicator that target is visible to missile (integer)

COMMON/MIFO

$$LM = \begin{cases} 1 \sim \text{target not visible to missile} \\ 0 \sim \text{otherwise} \end{cases}$$

18. SRLOCK = slant range (missile-target) when lock-on occurred (real, meters)

19. XPT = X-coordinate of last determined target location (real, meters)

20. YPT = Y-coordinate of last determined target location (real, meters)

21. ZPT = Z-coordinate of last determined target location (real, meters)

22. PC = fraction of target's height not visible to missile (real)

23. ANGL = the azimuth angle of the line of sight from an FO to a target (real, radians)

24. VC = the crossing velocity of a target relative to an FO (real, mils/sec)

25. SPR = slant range to the last determined target location (real, meters)

26. FOHIT = indicator whether the FO hit the target with a given pulse (integer)

$$FOHIT = \begin{cases} 1 \sim \text{FO hit the target} \\ 0 \sim \text{otherwise} \end{cases}$$

27. MIHIT = indicator whether missile has hit the target (integer)

$$MIHIT = \begin{cases} 1 \sim \text{missile hit the target} \\ 0 \sim \text{otherwise} \end{cases}$$

28. COVER = fraction of target covered with respect to missile (real)

29. PL = side limit of missile field of view (real)

INITIALIZATION: None

USERS: BFLITE, FINAL, FINALE, FLIGHT, PROB, PROBTV, PULSE, SEEKER, TVDET

LENGTH: 29

COMMON/MISAVE/

CONTENTS: MISAVE contains the parameters for the current indirect-fire missile element.

ARRANGEMENT: MISAVE(28,N)

where N is the number of indirect-fire missile elements.

All locations of COMMON/MISAVE/ are loaded internal to subroutine MICONG with data from common areas MIDATA, LNSET, and OPEN.

INITIALIZATION: None

USERS: MICONG, MICONP, FIRCON, MOV

LENGTH: 28 * N

COMMON/MISHIT/

CONTENTS: MISHIT sets impact parameters for the assessment of impacting missiles which did not have guidance to the target on impact.

ARRANGEMENT: MISHIT contains four parameters.

1. NFB = the artillery round equivalent to the missile upon impact. Artillery type NFB data will be used to assess the missile.

2. C1 = the coefficient of X^2 which describes the ellipse of effect of the impacting munition. The shape of the impacting area is $C1 \cdot X^2 + C2 \cdot XY + C3 \cdot Y^2$.

3. C2 = the coefficient of XY in the impact ellipse.

4. C3 = the coefficient of Y^2 in the impact ellipse.

INITIALIZATION: Required for all values.

USERS: FLIGHT

LENGTH: 4

COMMON/MISION/

CONTENTS: MISION describes the mission type of each maneuver unit. The codes are as follows:

0 if the maneuver unit is a helicopter team,
1 for attacking mission,
2 for delaying mission, and
3 for overwatch or fire support mission.

ARRANGEMENT: MISION is a one-dimensional array.

MISION(I) (integer)

I = 1,...,N maneuver unit numbers

where N is the total number of maneuver units.

INITIALIZATION: Required

USERS: ASSIGN, CASELE, FIRCON, FORM, INTEL, LOIMOV, MANOUT, MVCON

LENGTH: N

COMMON/MNTRIG/

CONTENTS: MNTRIG stores an array containing the firing threshold priority value required for on-call fires triggered by enemy elements for a given trigger area and fire support weapon code considered.

ARRANGEMENT: MNTRIG(J,K) (integer)

where

J = 1,...,30 trigger area number

K = 1,...,MWAIR fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 30 * MWAIR

COMMON/MPTR/

CONTENTS: MPTR contains the fire-request list entry number with whose requesting element an aerial team is negotiating a new mission; zero if no negotiation is underway.

ARRANGEMENT: MPTR is a one-dimensional array.

MPTR(I) (integer)

where

I = 1,...,NUMAVT aerial unit numbers.

INITIALIZATION: None

USERS: AIRFB

LENGTH: NUMAVT

COMMON/MREADY/

CONTENTS: MREADY contains the launcher firing preparation status.

ARRANGEMENT: MREADY is a one-dimensional array.

MREADY(I); I = 1,...,ITOTLN

$$= \begin{cases} 1 & \text{if launcher I has loaded missiles} \\ & \text{for an indirect-fire assignment} \\ 0 & \text{otherwise} \end{cases}$$

where I is the launcher number and ITOTLN is the number of launchers.

INITIALIZATION: None

USERS: MFB

LENGTH: ITOTLN

COMMON/NAMO/

CONTENTS: NAMO contains the current ammunition supply for an aerial vehicle.

ARRANGEMENT: NAMO is a two-dimensional array: (integer * 2)

NAMO(I,J) contains the supply of ammunition type I for the J^{th} aerial vehicle
for
I = 1,...,NAMMOS where NAMMOS is the maximum number of ammunition codes allowed per element
J = 1,...,MAXNAV where MAXNAV is the maximum number of aerial vehicle elements.

INITIALIZATION: None

USERS: AIRAMO, WASAIR

LENGTH: NAMMOS * MAXNAV, H

COMMON/NAVSEC/

CONTENTS: NAVSEC contains aerial section numbers corresponding to vehicle sections.

ARRANGEMENT: NAVSEC is a one-dimensional array.

NAVSEC(I) (integer)

$$\text{NAVSEC(I)} = \begin{cases} \text{N if section I is aerial section N} \\ \text{0 if section I is not an aerial section} \end{cases}$$

where I = 1,...,MNSEC section numbers.

INITIALIZATION: Required

USERS: DEFPOS,MISEND,SECPRM,COUNT,RTSRCH,RTCROS, RTSECT,RTLOIT,HXYMCP,RTATAK,DEFSET,HELFIR, DARFO,NATLDR,NEWMIS,FLGSET,HELCON,GETHEL, ATKPRM,HDIF,HELMOV,CBOBS,SECSET,OFFSET

LENGTH: MNSEC

COMMON/NAXIS/

CONTENTS: NAXIS contains the number of the current axis of advance for each maneuver unit.

ARRANGEMENT: NAXIS is a one-dimensional array.

NAXIS(I) (integer)

I = 1,...,N maneuver unit numbers

where N is the total number of maneuver units.

INITIALIZATION: Required as follows:

0	for helicopter units
1	for attacking maneuver units
J - 1	for delaying maneuver units
K - 1	for supporting fire maneuver units

where

J = the delay position occupied by a delaying maneuver unit at the start of the battle

K = the fire-support position occupied by a supporting fire maneuver unit at the start of the battle.

USERS: MVCON, LOIPOS, LOIMOV, SECSET

LENGTH: N

COMMON/NCBF/

CONTENTS: NCBF contains the number of entries in the counterbattery fire request list for each battery.

ARRANGEMENT: NCBF is a one-dimensional array.

NCBF(I) (integer)

where

NCBF(I) contains the number of entries for battery I for I = 1,...,N

where N is the number of batteries.

INITIALIZATION: Zero

USERS: AFB,AIC

LENGTH: N

COMMON/NCONC/

CONTENTS: NCONC stores an array containing the concentration number associated with a particular trigger area.

ARRANGEMENT: NCONC(I) (integer)

where I = 1,...,30 trigger area number

INITIALIZATION: Required

USERS: AFO,REVAL

LENGTH: 30

COMMON/NDELPT/

CONTENTS: NDELPT contains the number of the withdrawal point recorded for an aerial unit. Zero if no point has been determined.

ARRANGEMENT: NDELPT is a two-dimensional array.

NDELPT(I,J) (integer)

where

$$I = \begin{cases} 1 & \text{unit retirement position} \\ 2 & \text{unit defensive or waiting position, and} \end{cases}$$

J = 1,2,...,NUMAVT aerial unit numbers.

INITIALIZATION: None

USERS: DEFPOS,DEFSET

LENGTH: 2 * NUMAVT

COMMON/NFB/

CONTENTS: NFB stores an array containing the number of the firing battery, MISTIC launcher, or aerial team executing the mission of each forward observer and each ground-to-air communicator.

ARRANGEMENT: NFB(I) (integer)

where I = 1,...,ITOTFO FO number

$$I = \begin{cases} \text{ITOTFO} + 1 & \text{blue ground-to-air communicator} \\ \text{ITOTFO} + 2 & \text{red ground-to-air communicator.} \end{cases}$$

INITIALIZATION: Zero, altered during execution.

USERS: AFO,SELECA,AFSC,AIRFB,MFB

LENGTH: ITOTFO + 2

COMMON/NFLCRT/

CONTENTS: NFLCRT contains the minimum desired ratio of surviving MISTIC forward observers to surviving launchers in a unit with a specified MISTIC unit weapon code. The array is used to determine when MISTIC forward observer missions should be assigned to aerial units.

ARRANGEMENT: NFLCRT is a one-dimensional array.

NFLCRT(I) (integer)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: AFASGN

LENGTH: MWMIS - MWART

COMMON/NFMMAX/

CONTENTS: NFMMAX contains the maximum number of forward observers permitted to be assigned to a MISTIC unit of a specified weapon code. The array is used to limit the assignment of aerial units to MISTIC forward observer missions.

ARRANGEMENT: NFMMAX is a one-dimensional array.

NFMMAX(I) (integer)

where

I = 1,...,MWMIS - MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: AFASGN

LENGTH: MWMIS - MWART

COMMON/NFOCRT/

CONTENTS: NFOCRT contains the minimum desired ratio of forward observers to surviving artillery and aerial units per side. The array is used to determine when aerial units should be assigned special forward observer missions.

ARRANGEMENT: NFOCRT is a one-dimensional array.

NFOCRT(I) (integer)

where

$$I = \begin{cases} 1 & \text{indicates the blue force, and} \\ 2 & \text{indicates the red force.} \end{cases}$$

INITIALIZATION: Required

USERS: AFASGN

LENGTH: 2

COMMON/NFOFR/

CONTENTS: NFOFR stores an array containing the fire-support element who initiated fire request J on fire request net number K.

ARRANGEMENT: NFOFR(J,K) (integer)

where

J = 1,...,5 fire request list position number

K = 1,...,NTFRNT fire request net number.

INITIALIZATION: Zero, altered during execution.

USERS: AFO,AFDC,AFSC,FSCHEK,AIRFB,CBCONT,MFB

LENGTH: 5 * NTFRNT

COMMON/NFOMAX/

CONTENTS: NFOMAX contains the maximum number of forward observers allowed per side and is used to limit the assignment of aerial units to special forward observer missions.

ARRANGEMENT: NFOMAX is a one-dimensional array.

NFOMAX(I) (integer)

where

$$I = \begin{cases} 1 \text{ indicates the blue force, and} \\ 2 \text{ indicates the red force.} \end{cases}$$

INITIALIZATION: Required

USERS: AFASGN

LENGTH: 2

COMMON/NFRSAV/

CONTENTS: NFRSAV saves the maximum number of artillery fire requests that may be processed concurrently for each active Artillery Fire Direction Center. This common saves NFRMAX(I), when artillery unit I is assessed a period of inactivity as a result of counterbattery fire placed upon it.

ARRANGEMENT: NFRSAV(I) (integer)

where

I = 1,...,NUMART artillery unit.

INITIALIZATION: Zero, altered during execution.

USERS: BTLETH

LENGTH: NUMART

COMMON/NFUCRT/

CONTENTS: NFUCRT contains the maximum desired ratio of surviving MISTIC forward observers to surviving launchers in a unit with a specified MISTIC unit weapon code. The array is used to determine when MISTIC launcher missions should be assigned to aerial units.

ARRANGEMENT: NFUCRT is a one-dimensional array.

NFUCRT(I) (integer)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon code.

INITIALIZATION: Required

USERS: AFASGN

LENGTH: MWMIS-MWART

COMMON/NKPAT/

CONTENTS: NKPAT stores an array containing the number of fire patterns to be considered in selecting targets of opportunity for a given fire support weapon code.

ARRANGEMENT: NKPAT(I) (integer)

where

I = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: SELECA

LENGTH: MWART

COMMON/NLNMAX/

CONTENTS: NLNMAX contains the maximum number of launchers permitted to be assigned to a unit with a specified MISTIC unit weapon code. The array is used to limit the assignment of MISTIC launcher missions to aerial units.

ARRANGEMENT: NLNMAX is a one-dimensional array.

NLNMAX(I) (integer)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: AFASGN

LENGTH: MWMIS-MWART

COMMON/NM/

CONTENTS: NM contains the number of MISTIC missiles stowed but not loaded aboard a launcher.

ARRANGEMENT: NM is a one-dimensional array.

NM(J) (integer)

where

J = 1,...,ITOTLN launcher numbers.

INITIALIZATION: Required

USERS: LAUNCH, LOADM, MFB

LENGTH: ITOTLN

COMMON/NMLIM/

CONTENTS: NMLIM contains the maximum number of MISTIC missiles that may be loaded for fire from a launcher belonging to a unit with a specified MISTIC weapon code.

ARRANGEMENT: NMLIM(I) (integer)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: LOADM

LENGTH: MWMIS-MWART

COMMON/NOBVH/

CONTENTS: NOBVH stores an array containing the element number of the vehicle transporting each forward observer and each MISTIC launcher element. For forward observers, a zero indicates an FO Bravo.

ARRANGEMENT: NOBVH(I) (integer)

where

$$I = \begin{cases} 1,\ldots,\text{ITOTFO} & \text{FO number} \\ \text{ITOTFO} + 1,\ldots,\text{ITOTFO} + \text{ITOTLN} & \text{launcher number.} \end{cases}$$

INITIALIZATION: Required

USERS: AFO,DARFO,REASFO,AFSC,MFB,LOADM

LENGTH: ITOTFO + ITOTLN

COMMON/NOUTFG/

CONTENTS: NOUTFG contains outpost designators for delaying maneuver units.

ARRANGEMENT: NOUTFG is a linear array (integer).

NOUTFG(I) = number of the outpost occupied by delaying maneuver unit I

I = 1,2,...,N

where N is the number of maneuver units.

INITIALIZATION: Required. Entires for units not at outposts must be zero.

USERS: CNTOUT, MVCON, OTPT

LENGTH: N

COMMON/NPHAS/

CONTENTS: NPHAS contains values used by the movement controller model in representing phase lines.

ARRANGEMENT: NPHAS contains two values as follows:

NMPHAS = total number of phase lines represented (integer)

NXPHAS = the number of the phase line which the attackers are presently approaching (integer).

INITIALIZATION: Required. NXPHAS must be initialized to one.

USERS: MVCON

LENGTH: 2

COMMON/NRDCNT/

CONTENTS: COMMON/NRDCNT/ contains the number of rounds fired by a specified CSW team at its current firing position.

ARRANGEMENT: NRDCNT is a one-dimensional array:

NRDCNT(NCR)

NCR = 1,...,NCSWMX
where NCSWMX is the maximum number of CSW teams.

INITIALIZATION: Zero, altered during execution

USERS: CSWCON

LENGTH: NCSWMX

COMMON/NREQR/

CONTENTS: NREQR contains the minimum allowed number of surviving elements in an aerial section with a specified aerial weapon code. The section will retire if this value is violated.

ARRANGEMENT: NREQR is a one-dimensional array.

NREQR(I) (integer)

where

I = 1,...,N aerial weapon codes
(See COMMON/ANGLIM/ for a definition of weapon code.

INITIALIZATION: Required

USERS: RETIRE,FUELD

LENGTH: N

COMMON/NRL/

CONTENTS: NRL contains the number of missiles loaded per loading operation on a launcher belonging to a unit with a specified MISTIC weapon code.

ARRANGEMENT: NRL(I) (integer)

where

I = 1,...,MWMIS-MWART.

INITIALIZATION: Required

USERS: LOADM,LAUNCH

LENGTH: MWMIS-MWART

COMMON/NRNDAT/

CONTENTS: NRNDAT stores an array containing the number of rounds required for the search and destroy missions against concentration area I.

ARRANGEMENT: NRNDAT is a one-dimensional array:

NRNDAT(I) (integer)

where I = 1,...,20 search and destroy concentration areas

INITIALIZATION: Required

USERS: FSCHEK

LENGTH: 20

COMMON/NRNDCN/

CONTENTS: NRNDCN stores an array containing the number of rounds to be fired in a specified concentration area by a specified fire support weapon code when fire is triggered by movement of friendly elements.

ARRANGEMENT: NRNDCN(J,K) (integer)

where

J = 1,...,20 concentration number

K = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 20 * MWART

COMMON/NSCHCN/

CONTENTS: NSCHCN contains the number of the planned concentration area associated with a scheduled fire for an artillery firing battery (see common areas XCONC,YCONC).

ARRANGEMENT: NSCHCN is a two-dimensional array.

NSCHCN(I,J) (integer)

NSCHCN(I,J) contains the concentration number of the I^{th} fire for the J^{th} battery for $I = 1,...,6; J = 1,...,N$

where N is the number of batteries.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: 6 * N

COMMON/NSTHFF/

CONTENTS: NSTHFF stores an array containing the fire fight flag for each forward observer.

ARRANGEMENT: NSTHFF is a one-dimensional array (integer)

$$NSTHFF(I) = \begin{cases} 1 & \text{indicates a fire fight exists} \\ 0 & \text{otherwise} \end{cases}$$

for I = 1,...,ITOTFO, FO numbers,

where ITOTFO is the total number of FO's.

INITIALIZATION: Zero, altered during execution

USERS: AFO, AFSC, ARFO, HELFIR, NEWFO, WRITFO

LENGTH: ITOTFO

COMMON/NTELE/

CONTENTS: NTELE contains clock location data for the fire support model.

ARRANGEMENT: NTELE contains four values (integer).

1. NTFO contains the clock number of the last missile clock.
2. NTFDC contains the clock number of the last forward observer clock.
3. NTFB contains the clock number of the last fire direction center and radio net.
4. NAIC contains the clock number of the last fire-support firer.

INITIALIZATION: Required

USERS: AFB,AFDC,AFO,ARFO,CASELE,FIRCON,SEQCNT, SEQSET,AFDC,AFSC,FSCHEK,SECSET,MFB

LENGTH: 4

COMMON/NTOBAL/

CONTENTS: NTOBAL contains the number of aerial vehicle teams allowed over the battlefield.

ARRANGEMENT: NTOBAL(I) = maximum number of aerial vehicle teams allowed over the battlefield at one time (integer)

where

$$I = \begin{cases} 1 & \text{for blue force, and} \\ 2 & \text{for red force.} \end{cases}$$

INITIALIZATION: Required

USERS: AVAIL

LENGTH: 2

COMMON/NTOT/

CONTENTS: Common NTOT contains the number of crews mounted or assigned to an APC carrier.

ARRANGEMENT: NTOT is a one-dimensional array:

NTOT(I) (integer)

where I = 1,...,NAPCMX APC carrier

INITIALIZATION: Required

USERS: CSWCON, MOUNT

LENGTH: NAPCMX

COMMON/NTRANS/

CONTENTS: NTRANS stores an array containing the radio net number over which each forward observer and ground-to-air communicator is currently transmitting.

ARRANGEMENT: NTRANS(I) (integer)

where

$$I = \begin{cases} 1,\ldots,\text{ITOTFO} & \text{FO number} \\ \text{ITOTFO} + 1 & \text{blue ground-to-air communicator} \\ \text{ITOTFO} + 2 & \text{red ground-to-air communicator.} \end{cases}$$

and ITOTFO is the total number of FO's.

INITIALIZATION: Zero, altered during execution.

USERS: AFO, AFSC, FSCMON, WRITFO

LENGTH: ITOTFO + 2

COMMON/NUMFIR/

CONTENTS: NUMFIR stores an array containing the number of the fire support firer to which a message is addressed on each radio net.

ARRANGEMENT: NUMFIR(I) (integer)

where

I = 1,...,NTFRNT fire request net.

INITIALIZATION: Zero, altered during execution.

USERS: AFDC,AFO

LENGTH: NTFRNT

COMMON/NUMSEN/

CONTENTS: NUMSEN contains the number of senders on each net queue.

ARRANGEMENT: NUMSEN is a one-dimensional array.

NUMSEN(I) (integer)

I = 1,...,NPLT platoon net subscripts
I = NPLT + 1,..., NPLT + NTEM company net subscripts
I = NPLT + NTEM + 1 blue battalion net subscript
I = NPLT + NTEM + 2 red battalion net subscript

where

NPLT = the total number of platoons
NTEM = the total number of teams

INITIALIZATION: None

USERS: QUE, QUECHK, QUEP, SENDC

LENGTH: NTEM + NPLT + 2

COMMON/NUMBER/

CONTENTS: NUMBER contains organization parameters.

ARRANGEMENT: NUMBER contains 39 values (integer).

ORDER LISTED	MNEMONIC	DEFINITION
1	0	normal zero
2	NUMBLU	number of blue elements
3	NUMELE	total number of elements
4	NBSEC	number of blue sections
5	NSEC	total number of sections
6	NBPLT	number of blue platoons
7	NPLT	total number of platoons
8	NBTEM	number of blue teams
9	NTEM	total number of teams
10	NBMAN	number of blue maneuver units
11	NMAN	total number of maneuver units
12	NUMRED	number of red elements

13	INFINITY	largest integer number
(13)	FINITY	largest floating-point number
14	NBLUFO	number of blue FO's
15	ITOTFO	total number of FO's
16	NBLULN	number of blue launchers
17	ITOTLN	total number of launchers
18	NAMMOS	number of ammunition codes used
19	NOTPST	number of outposts represented
20	NBART	number of blue artillery units
21	NUMART	total number of artillery units
22	MNSYS	total number of weapon systems
23	MNFRT	total number of direct-fire weapon codes
24	NBMIS	number of blue MISTIC units
25	MISTUN	total number of MISTIC units
26	NBAVT	number of blue aerial vehicle teams
27	NUMAVT	total number of aerial vehicle teams
28	NTSFO	FO number of the last special forward observer
29*	MWBART	maximum weapon code assigned to blue artillery units
30*	MWART	maximum weapon code assigned to artillery units
31*	MWBMIS	maximum weapon code assigned to blue MISTIC units
32*	MWMIS	maximum weapon code assigned to MISTIC units
33*	MWBAIR	maximum weapon code assigned to blue aerial teams
34*	MWAIR	maximum weapon code assigned to aerial teams
35	NTFRNT	total number of fire request nets (or receivers)
36	NTFSFT	total number of fire support firing teams
37	NBAIC	number of blue artillery intelligence centers
38	NUMAIC	total artillery intelligence centers
39	MAXLWC	the maximum value of weapon code appearing for vehicular elements, which are not helicopter elements

*Note: Fire support weapon codes are arranged in monotonically increasing order starting at one, such that:
$0 \leq \text{MWBART} \leq \text{MWART} \leq \text{MWBMIS} \leq \text{MWMIS} \leq \text{MWBAIR} \leq \text{MWAIR}$

INITIALIZATION: Required

USERS: Used throughout the simulation.

LENGTH: 39

COMMON/NUMTRG/

CONTENTS: NUMTRG contains the trigger number which caused an on-call fire to be requested by a forward observer.

ARRANGEMENT: NUMTRG(I) (integer)

where

I = 1,...,ITOTFO FO number

INITIALIZATION: Zero, altered during execution.

USERS: AFO

LENGTH: ITOTFO

COMMON/NXCONC/

CONTENTS: NXCONC stores an array containing the number of the next concentration area to be fired upon following completion of fires upon concentration I.

ARRANGEMENT: NXCONC(I) (integer)

where

I = 1,...,20 concentration number

INITIALIZATION: Required

USERS: AFO

LENGTH: 20

COMMON/NXMIS/

CONTENTS: NXMIS contains the number of the next planned concentration for an artillery firing battery to fire at the conclusion of the present area fire in the series.

ARRANGEMENT: NXMIS is a one-dimensional array.

NXMIS(I) (integer)

where

NXMIS(I) is the next concentration for the I^{th} battery

for I = 1,...,N

where N is the number of batteries.

INITIALIZATION: Zero

USERS: AFB, WRTTRG

LENGTH: N

COMMON/ONDIST/

CONTENTS: ONDIST stores an array containing the effective radius of concentration area J used to compute on-call target duration time for fire support weapon code K.

ARRANGEMENT: ONDIST(J,K) (real, meters)

where

J = 1,...,20 concentration area

K = 1,...,MWAIR fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 20 * MWAIR

COMMON/OPEN/

CONTENTS: OPEN contains information associated with the missile flight procedure.

ARRANGEMENT: OPEN contains 35 values:

1. J = indicator of missiles flight progress: subscript of common arrays RM, ZM, ZMD, TF, and TD (integer)

2. FL = forward limit of missile's field of view (real, meters)

3. ITIMES = target's element number (dummy variable) (integer)

4. Z1 = FO's target location error perpendicular to line of sight (real, mils)

5. Z2 = FO's target location error along line of sight (real, mils)

6. K = indicator of first iteration (integer)

$$K = \begin{cases} 0 \sim \text{first iteration} \\ 1 \sim \text{otherwise} \end{cases}$$

7. N = number of consecutive missed pulses (integer)

8. RM = missile's horizontal flight range from launch point (real, meters)

9. ANGM = missile's heading (real, radians)

10. SINM = sine of ANGM (real)

11. COSM = cosine of ANGM (real)

12. L = indicator of target location within missile's field of view (integer)

$$L = \begin{cases} 1 \sim \text{target located within fringe region} \\ 0 \sim \text{target location within center region} \end{cases}$$

13. DD = distance traveled by the missile during one illuminator pulse (real, meters)

14. NH = indicator of missile pulse detection

$$NH = \begin{cases} 1 \sim \text{this pulse was missed} \\ 0 \sim \text{otherwise} \end{cases}$$

15. ANGFO = FO's direction of travel (real, radians)

16. SANGFO = sine of ANGFO (real)

17. CANGFO = cosine of ANGFO (real)

18. TT = target's battle clock time (real, seconds)

19. TFO = FO's battle clock time (real, seconds)

20. FLAG = indicator whether FO is target (logical)

$$FLAG = \begin{cases} \text{TRUE} \sim \text{FO is target} \\ \text{FALSE} \sim \text{otherwise} \end{cases}$$

21. DT = time increment for each pulse (real, seconds)

22. SR = *slant range, missile to target (real, meters)*

23. DYP = distance from target to the fringe region limit of the field of view (real, meters)

24. DPL = width of fringe region (real, meters)

25. LFON = target's FO (FO number) (integer)

26. LOCK = indicator whether acquisition has occurred (integer)

$$LOCK = \begin{cases} 1 \sim \text{acquisition has occurred} \\ 0 \sim \text{otherwise} \end{cases}$$

27. TMC = dummy variable

28. AFO = average acceleration for FO over his last event (real, meters/sec^2)

29. AT = average acceleration for target over his last event (real, meters/sec^2)

30. VFT = target's final velocity at the end of his last event (real, meters/second)

31. VFFO = FO's final velocity at the end of his last event (real, meters/second)

32. XAT = target's X-coordinate at end of his last event (real, meters)

33. YAT = target's Y-coordinate at end of his last event (real, meters)

34. XAF = FO's X-coordinate at end of his last event (real, meters)

35. YAF = FO's Y-coordinate at end of his last event (real, meters)

INITIALIZATION: None

USERS: FLIGHT, FINAL, MICONG

LENGTH: 35

COMMON/ORESPN/

CONTENTS: ORESPN stores an array containing the average reaction time of a fire support unit with weapon code I for on-call targets.

ARRANGEMENT: ORESPN(I) (real, seconds)

where

I = 1,...,MWART artillery unit weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: MWART

COMMON/OUTPST/

CONTENTS: OUTPST contains four linear arrays describing characteristics of outposts.

ARRANGEMENT:

$$IWDRAW(I) = \begin{cases} 1 & \text{if units have begun to withdraw} \\ 0 & \text{if otherwise (integer)} \end{cases}$$

OTMDEL(I) = minimum delay time required for withdrawal (real, seconds)

OUTTBP(I) = minimum threat level required for withdrawal (real)

OUTRBP(I) = minimum range to enemy force required for withdrawal (real, meters)

I = 1,...,N,...,10

where N is the number of outposts (N ≤ 10).

INITIALIZATION: Required for OTMDEL,OUTTBP and OUTRBP. IWDRAW must be initialized with zeros.

USERS: MVCON

LENGTH: 4 * 10

COMMON/PALLOW/

CONTENTS: Common area PALLOW contains limits of visibility used in firer-target selection, fire position selection and missile launching.

ARRANGEMENT: PALLOW contains one constant.

1. Maximum allowable percentage of the target height which can be covered relative to forward observer, or a firer moving to a fire position.

INITIALIZATION: Required

USERS: FIRCON, MOVE

LENGTH: 1

COMMON/PHNG

CONTENTS: PHNG contains the probability of a successful direct-fire launch of a MISTIC missile from a helicopter having a specified helicopter section weapon code.

ARRANGEMENT: PHNG(I) (real)

where

I = 1, ..., N helicopter section weapon codes.

INITIALIZATION: Required

USERS: HLNCH

LENGTH: N

COMMON/PKART/

CONTENTS: PKART contains kill probabilities for artillery fire against ground elements.

ARRANGEMENT: PKART is a three-dimensional array.

PKART(I,J,K) (real)

where

PKART(I,J,K) contains the kill probability of kill type I by battery J against weapon code K

for I = 1,...,4
J = 1,...,8
K = 1,...,N where N is the maximum weapon code.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: 4*8*N

COMMON/PKPB/

CONTENTS: PKPB contains kill probabilities and fixed biases for each firing event for printing. Kill types not considered (i.e., a lesser kill type occurred) will be zero.

ARRANGEMENT: PKPB contains seven variables as follows:

1. PNH - probability of no kill
2. PMH - probability of a mobility kill
3. PFH - probability of a firepower kill
4. PMFH - probability of a mobility firepower kill
5. PKH - probability of a complete kill

6. XTOT - horizontal fixed bias

7. YTOT - vertical fixed bias.

INITIALIZATION: None

USERS: FIRMOD, HPROB, TLETH

LENGTH: 7

COMMON/PKTNK/

CONTENTS: PKTNK contains sets of kill probabilities corresponding to tank type targets.

ARRANGEMENT: PKTNK is a five-dimensional array.

PKTNK(I,J,K,L,M) (real)

I = 1,...,6	subscripts corresponding to range intervals as defined in COMMON/HPRNG/
J = 1,...,5	subscripts corresponding to kill types
K = 1,...,7	subscripts corresponding to firer-target aspect angle as defined in COMMON/TARASP/
L = 1,...,6	subscripts corresponding to target speeds as defined in COMMON/TARSPD/
M = 1,..., MTNKLT	subscript corresponding to firer-ammo-target combinations (e.g., KCHAR from COMMON/IAMMO/) where MTNKLT is the maximum value of KCHAR for tank-target combinations.

INITIALIZATION: Required.

USERS: DDSX4

LENGTH: 6·5·7·6·MTNKLT

COMMON/PNG/

CONTENTS: PNG contains the probability of a successful launch from a launcher belonging to a unit with specified MISTIC weapon code.

ARRANGEMENT: PNG(I) (real)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB, LAUNCH

LENGTH: MWMIS-MWART

COMMON/RADINC/

CONTENTS: RADINC contains the incremental reduction in the enemy complex circled, described by COMMON/RADMAX/, for use in selecting a target complex by an aerial vehicle section currently having available a given number of suppressive fire and point fire weapons.

ARRANGEMENT: RADINC is a two-dimensional array.

RADINC(I,J) contains the increment for an air section having I-1 suppressive fire and J-1 point fire weapons available.

for

I = 1,...,NAFW+1 where NAFW is the maximum number of suppressive fire weapons available

J = 1,...,NPFW+1 where NPFW is the maximum number of point fire weapons for a section.

INITIALIZATION: Required

USERS: AIRFIR

LENGTH: (NAFW+1)*(NPFW+1)

COMMON/RADMA/

CONTENTS: RADMA contains the radii of mission operations areas for aerial units of a given force flying missions of specified types.

ARRANGEMENT: RADMA(I,J) (real, meters)

where

$$J = \begin{cases} 1 \text{ for blue force} \\ 2 \text{ for red force} \end{cases}$$

$$I = \begin{cases} 1 \text{ indicates a standby mode} \\ 2 \text{ indicates a direct-fire attack mission} \\ 3 \text{ indicates an observation mission} \\ 4 \text{ indicates an indirect-fire mission.} \end{cases}$$

INITIALIZATION: Required

USERS: RTCROS,RTSRCH

LENGTH: 4 * 2

COMMON/RADMAX/

CONTENTS: RADMAX contains the maximum allowable radius within which a ground target complex will be attacked by an aerial vehicle section currently having available a given number of suppression fire and point fire weapons.

ARRANGEMENT: RADMAX is a two-dimensional array.

RADMAX(I,J) contains the radius for an air section having I-1 suppressive fire and J-1 point fire weapons available

for I = 1,...,NAFW + 1 where NAFW is the maximum number of suppressive fire weapons for a section

J = 1,...,NPFW + 1 where NPFW is the maximum number of point fire weapons for a section.

INITIALIZATION: Required

USERS: AIRFIR

LENGTH: (NAFW + 1)(NPFW + 1)

COMMON/RAFMNF/

CONTENTS: RAFMNF contains the adjusted target range beyond which a given type of forward observer will not engage a target with direct fire.

ARRANGEMENT: RAFMNF is a two-dimensional array.

RAFMNF(I,J) (real, meters)

where

$$I = \begin{cases} 1 & \text{indicates an artillery FO} \\ 2 & \text{indicates a MISTIC FO} \\ 3 & \text{indicates a special FO, and} \end{cases}$$

$$J = \begin{cases} 1 & \text{indicates the FO belongs to the blue force} \\ 2 & \text{indicates the FO belongs to the red force.} \end{cases}$$

INITIALIZATION: Required

USERS: FIRCON

LENGTH: 3 * 2

COMMON/RAFMNL/

CONTENTS: RAFMNL contains the adjusted target range beyond which a launcher having a specified weapon code will not engage a target with direct fire.

ARRANGEMENT: RAFMNL is a one-dimensional array.

RAFMNL(I) (real, meters)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: FIRCON

LENGTH: MWMIS-MWART

COMMON/RAM/

CONTENTS: RAM stores an array containing the maximum target engagement range for a specified artillery or MISTIC weapon code.

ARRANGEMENT: RAM(I) (real, meters)

where

I = 1,...,MWMIS fire support weapon code.

INITIALIZATION: Required

USERS: AVAIL

LENGTH: MWMIS

COMMON/RATT/

CONTENTS: RATT contains the range adjustment factor for a specified firer weapon-ammunition code, target weapon system code.

ARRANGEMENT: RATT is a two-dimensional array.

RATT(J,KF)

KF = 1,...,NF the number of firer weapon-ammunition codes from COMMON/IHTPRB/

J = 1,...,NT the number of weapon system codes.

INITIALIZATION: Required

USERS: ADTGSL, FIRCON

LENGTH: NT * NF

COMMON/RDFMIN/

CONTENTS: RDFMIN contains the minimum range from which a weapon of a specified type may be launched from an aerial section with specified aerial weapon code.

ARRANGEMENT: RDFMIN(I,J) (real, meters)

where

$$I = \begin{cases} 1 & \text{indicates direct-fire MISTIC missile} \\ 2 & \text{indicates a beam-rider missile} \\ 3 & \text{other point-fire weapons} \\ 4 & \text{suppressive fire weapons} \end{cases}$$

J = 1,...,N helicopter weapon code numbers (See COMMON/ANGLIM/ for definition of weapon code).

INITIALIZATION: Required

USERS: RTATAK

LENGTH: 4 * N

COMMON/REACT/

CONTENTS: REACT contains time required for an aerial unit to move from its loiter position off the battlefield to its initial battlefield position.

ARRANGEMENT: REACT(I) (real, seconds)

where

I = 1,2,...,NUMAVT aerial unit numbers.

INITIALIZATION: Required

USERS: SECSET

LENGTH: NUMAVT

COMMON/REDDET/

CONTENTS: REDDET contains the detection codes for each red element: a 0 for no knowledge, a 1 for approximate knowledge, a 2 for full visual detection, and a 3 for a pinpoint detection.

ARRANGEMENT: REDDET is a two-dimensional array of halfwords:

REDDET (I,J) (integer * 2)

I = 1,...,NUMBLU the blue element numbers
J = 1,..., NUMELE - NUMBLU the red element numbers reduced to the range 1,..., NRED, where NUMBLU is the total number of blue elements.

NRED is the total number of red elements, and NUMELE is the total number of elements.

INITIALIZATION: None

USERS: AIRFIR, AIRPOR, GETDET, IDET, IDETP, KOMDET

LENGTH: NUMBLU * NRED halfwords = (NUMBLU * NRED),H

SCENARIO DEPENDENT

Routines using Common /REDDET/ may require recompilation to obtain the desired maximum value for subscript I.

COMMON/REMONT/

CONTENTS: Common /REMONT/ contains the remount time for a CSW unit that has reached its carrier, following its return from a firing position.

ARRANGEMENT: REMONT is a one-dimensional array:

REMONT(NKWEP) (real, seconds)

for

NKWEP = 1,...,NCWMAX CSW unit's weapon code

where NCWMAX is the total number of CSW weapon codes.

INITIALIZATION: Required

USERS: MOUNT

LENGTH: NCWMAX

COMMON/RETFCN/

CONTENTS: RETFCN contains the fraction of an aerial unit that must retire before the unit itself will retire, given the unit has a specified aerial weapon code.

ARRANGEMENT: RETFCN(I) (real)

where

I = 1,2,...,MWAIR-MWMIS aerial unit weapon codes.

INITIALIZATION: Required

USERS: ATDEC

LENGTH: MWAIR-MWMIS

COMMON/RF/

CONTENTS: RF contains target surface average reflectance.

ARRANGEMENT: RF is a one-dimensional array.

RF(I) = target surface average reflectance (real, foot-lamberts/lumen/feet^2)

where

I = 1,...,N target weapon system code numbers.

INITIALIZATION: Required

USERS: DETH

LENGTH: N

COMMON/RFOMAX/

CONTENTS: RFOMAX contains an array that specifies the range from the target at which initiation of illumination is desired for aerial sections illuminating targets for indirect fire MISTIC attack.

ARRANGEMENT: RFOMAX is a one-dimensional array.

RFOMAX(I) (real, meters)

where

I = MISTIC weapon code of unit to which aerial section is assigned.

INITIALIZATION: Required

USERS: RTATAK

LENGTH: MWMIS-MWART

COMMON/RFUEL/

CONTENTS: RFUEL contains the rate at which fuel is expended by aerial elements having specified aerial weapon codes and performing specified movement activities.

ARRANGEMENT: RFUEL(I,J) (real, pounds/second)

where

$$I = \begin{cases} 1 & \text{when aerial section is loitering} \\ 2 & \text{when aerial section is performing enroute movement} \\ 3 & \text{when aerial section is attacking a target} \\ 4 & \text{when aerial section is searching for a target} \end{cases}$$

J = 1,...,N aerial weapon codes (See COMMON/ANGLIM/).

INITIALIZATION: Required

USERS: SECSET,HELMOV

LENGTH: 4 * N

COMMON/RGPTS/

CONTENTS: RGPTS contains the probabilities of beam acquisition at the slant ranges given in COMMON/SRPTS.

ARRANGEMENT: RGPTS is a one-dimensional array.

RGPTS(I),

I = 1, ···, N

where RGPTS(I) is the probability that the beam is acquired at slant range SRPTS(I), and

N = number of distance points.

INITIALIZATION: Required

USERS: FINAL

LENGTH: N

COMMON/RLNMAX/

CONTENTS: RLNMAX contains an array that specifies the range from the target at which an indirect-fire MISTIC launch sequence of an aerial section should commence.

ARRANGEMENT: RLNMAX is a one-dimensional array.

RLNMAX(I) (real, meters)

where

I = MISTIC weapon code of unit to which aerial section is assigned.

INITIALIZATION: Required

USERS: RTATAK

LENGTH: MWMIS-MWART

COMMON/RM/

CONTENTS: RM contains the down-range missile distance at corresponding time TF (given in COMMON/TF).

ARRANGEMENT: RM(I) (real, meters)

$I = 1, \cdots, N$

where N = number of distance points,

RM(1) = 0.0

INITIALIZATION: Required

USERS: FLIGHT

LENGTH: N

COMMON/RMAX

CONTENTS: Common RMAX contains the maximum desirable firing range for a CSW unit of a given weapon code.

ARRANGEMENT: RMAX is a one-dimensional array:

RMAX (I) (real, meters)

I = 1,...,NCWMAX CSW weapon code

where NCWMAX is the maximum number of weapon codes for crew served weapons.

INITIALIZATION: Required

USERS: CSWGRD

LENGTH: NCWMAX

COMMON/RMIN/

CONTENTS: Common RMIN consists of a single variable and an array containing the minimum desirable firing range for a CSW unit of a given weapon code.

ARRANGEMENT:

1) RMINO - the minimum distance between a threat position and an APC to allow an APC to dismount crews in the attack mode.

2) RMIN(I) - the minimum desired firing range for a CSW unit of weapon code I. (real, meters)

I = 1,...,NCWMAX CSW weapon code

where NCWMAX is the maximum number of weapon codes for crew served weapons.

INITIALIZATION: Required

USERS: CSWCON, CSWGRD

LENGTH: NCWMAX + 1

COMMON/RSTAS/

CONTENTS: RSTAS contains the radius of a semicircle defining the end of a loiter station of a specified type occupied by an aerial section with a given aerial weapon code.

ARRANGEMENT: RSTAS(I,J) (real, meters)

where

I =
- 1 indicates section is waiting for fire support
- 2 indicates section is waiting at retirement position
- 3 indicates section is waiting at defensive position.

J = 1,...,N aerial section weapon codes (See COMMON/ANGLIM/).

INITIALIZATION: Required

USERS: RTLOIT

LENGTH: 3 * N

COMMON/RSTAU/

CONTENTS: RSTAU contains the radius of a semicircle defining the end of a loiter station of a given type occupied by an aerial unit with a specified weapon code.

ARRANGEMENT: RSTAU(I,J) (real, meters)

where

I =
- 1 indicates unit is awaiting a mission
- 2 indicates unit is at retirement position
- 3 indicates unit is at defensive position
- 4 indicates unit is loitering as an indirect-fire support unit, and

J = 1,...,MWAIR-MWMIS aerial unit weapon codes.

INITIALIZATION: Required

USERS: RTLOIT

LENGTH: 4 * (MWAIR-MWMIS)

COMMON/RTKONH/

CONTENTS: RTKONH contains constants used in dynamic route selection for aerial sections and units.

ARRANGEMENT: RTKONH contains six values:

1. ISTRT = route selection grid row containing the initial point of an aerial vehicle section route (ISTRT must be even and not greater than IMAX (see COMMON/TACKON/)).

2. JSTRT = route selection grid column containing the initial point of an aerial vehicle section route (JSTRT must be equal to JFIN and satisfy the relation $1 \leq JSTRT \leq 5$).

3. EMIN = the distance between rows of an aerial vehicle route selection grid.

4. IFIN = route selection grid row containing the final point of an aerial vehicle section route (IFIN must be even and must satisfy the relation $ISTRT \leq IFIN \leq IMAX$ (see COMMON/TACKON/)).

5. JFIN = route selection grid column containing the final point of an aerial vehicle section route (JFIN must equal JSTRT).

6. RADIUS = distance from the initial point in the E(I,J) grid within which the computed route is used.

INITIALIZATION: Required

USERS: RTSELH, XYLOCH

LENGTH: 6

COMMON/RTRIG/

CONTENTS: RTRIG stores an array containing the radius of each trigger area.

ARRANGEMENT: RTRIG(I) (real, meters)

where

I = 1,...,30 trigger area number.

INITIALIZATION: Required

USERS: REVAL,TRIG

LENGTH: 30

COMMON/RXFIRE/

CONTENTS: Common RXFIRE contains the distance to the threat position at the Kth point defining the range fire power index versus distance curve.

ARRANGEMENT: RXFIRE is a one-dimensional array:

RXFIRE(K) (real, meters)

K = 1,...,ICRMAX

where ICRMAX is the number of points in the array, a user supplied storage dimension parameter.

INITIALIZATION: Required

USERS: CSWDES

LENGTH: ICRMAX

COMMON/RYFIRE/

CONTENTS: Common RYFIRE contains the value of the range fire power index for a CSW unit with weapon code NKWEP corresponding to a distance RXFIRE(K).

ARRANGEMENT: RYFIRE(K,NKWEP) (real)

K = 1,...,ICRMAX
NKWEP = 1,...,NCWMAX CSW weapon code

where ICRMAX is the number of points used in COMMON/RXFIRE/ and NCWMAX is the maximum number of weapon codes for crew served weapons.

INITIALIZATION: Required

USERS: CSWDES

LENGTH: ICRMAX * NCWMAX

SCENARIO DEPENDENT

Routines using COMMON/RYFIRE/ may need to be recompiled to obtain the desired maximum value for subscript K.

COMMON/SATARG/

CONTENTS: SATARG contains speed and aspect data for firer and target for each firing event.

ARRANGEMENT: SATARG contains thirteen values as follows:

1. ASPCT1 = the firer-target aspect angle for computing hit probability (real, radians)

2. ASPCT2 = the firer-target aspect angle for computing kill probabilities (real, radians)

3. AFACT1 = the interpolation factor for the hit probability aspect angle (real)

4. AFACT2 = the interpolation factor for the kill probability aspect angle (real)

5. IA1 = the subscript of the interval in COMMON/TARASP/ which contains the hit probability aspect angle (integer)

6. IA2 = the subscript of the interval in COMMON/TARASP/which contains the kill probability aspect angle (integer)

7. SPDFC1 = the interpolation factor for the target speed (real)

8. SPDFC2 = the interpolation factor for the firer speed (real)

9. IS1 = the subscript of the interval in COMMON/TARSPD/which contains the target speed (integer)

10. IS2 = the subscript of the interval in COMMON/VEHSPD/which contains the firer speed (integer)

11. RNGFAC = the interpolation factor for the firer-target range (real)

12. IR = the subscript of the interval in COMMON/HPRNG/ which contains the firer-target range (integer)

13. ASPECT = the true fire target aspect angle (real, radians)

INITIALIZATION: None

USERS: DDSX2, DDSX3, DDSX4, FIRMOD, SPDASP

LENGTH: 13

COMMON/SC/

CONTENTS: Common SC contains the x-coordinate of both red and blue force strongpoints.

ARRANGEMENT: SC is a one-dimensional array:

SC(I)

I = 1,...,NRSP   subscripts for the red strongpoints

I = NRSP + 1,...,NRSP + NBSP   subscripts for the blue strongpoints

where NBSP is the number of red strongpoints and NBSP is the number of blue strongpoints, user-supplied storage dimension parameters.

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: NRSP + NBSP

COMMON/SCANNS/

CONTENTS: SCANNS contains constants used in defining the field of view for an indirect-fire guided missile using a ballistic trajectory.

ARRANGEMENT: SCANNS contains seven values arranged as follows:

1. SENANG = the angle in a vertical plane containing the missile direction of flight between the normal to the missile direction of flight and the axis of the field of view (real, radians).

2. ALPHAR = the angle in a vertical plane containing the missile direction of flight between the axis of the field of view and rear edge of the field of view (real, radians).

3. ALPHAF = the angle in a vertical plane containing the missile direction of flight between the axis of the field of view and the forward edge of the field of view (real, radians).

4. BETAL = the angle in a plane containing the sensor and the rear edge of the trapezoid between: 1) a normal to the rear edge of the trapezoid through the sensor, and 2) the intersection between the rear edge of the trapezoid and the left lateral side of the trapezoid (real, radians).

5. BETAR = the analogue of BETAL for the right side of the trapezoid.

6. DELTAL = the angle in a plane containing the sensor and the front edge of the trapezoid between: 1) a normal to the front edge of the trapezoid through the sensor, and 2) the intersection between the front edge of the trapezoid and the left lateral side of the trapezoid (real, radians).

7. DELTAR = the analogue of DELTAL for the right side of the trapezoid.

INITIALIZATION: Required.

USERS: SEEKER

LENGTH: 7

COMMON/SCAPH/

CONTENTS: SCAPH corresponds exactly to SCAP but is associated with estimates of difficulty prepared by a helicopter section leader.

ARRANGEMENT: See COMMON/SCAP/.

INITIALIZATION: Required

USERS: RTSELH

LENGTH: NRWP + NBWP

COMMON/SCHTIM/

CONTENTS: SCHTIM contains the desired start time for an artillery scheduled fire mission.

ARRANGEMENT: SCHTIM is a two-dimensional array.

SCHTIM(I,J) (real, seconds)

SCHTIM(I,J) contains the time for the I^{th} scheduled fire by the J^{th} battery for I = 1,...,6; J = 1,...,N

where N is the number of batteries.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: 6 * N

COMMON/SDL/

CONTENTS: SDL contains the standard deviation of time distribution whose mean is TBL.

ARRANGEMENT: SDL(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: LOADM, LAUNCH

LENGTH: MWMIS-MWART

COMMON/SFR/

CONTENTS: SFR contains the standard deviation of the time distribution whose mean is TBFR.

ARRANGEMENT: SFR(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB,LAUNCH

LENGTH: MWMIS-MWART

COMMON/SHBM/

CONTENTS: SHBM contains the standard deviation of the distribution of time required to launch a direct fire missile of a given type from a helicopter having a given aerial weapon code.

ARRANGEMENT: SHBM(I,J) (real, seconds)

where I = { 1 for a MISTIC missile
2 for a beam-rider missile, and

J = 1,...,NA3, the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON /LWCOD/ are numbered 1,...,MAXLWC for ground elements and from MAXLWC + 1,...,MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: TFCOMP

LENGTH: 2 * NA3

COMMON/SIGADJ/

CONTENTS: SIGADJ stores an array containing the standard deviations of the distribution of time required to adjust firing data after receipt of an adjustment message over a specified radio net.

ARRANGEMENT: SIGADJ is a one-dimensional array:

SIGADJ(I) (real, seconds)

I = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/SIGAL/

CONTENTS: SIGAL contains the standard deviation of the distribution of time for an artillery firing battery to load, lay, and fire the initial volley of a fire mission of specified weapon code (see COMMON/UAL/).

ARRANGEMENT: SIGAL is a one-dimensional array:

SIGAL(I) (real, seconds)

SIGAL(I) contains the time for the I^{th} artillery fire support weapon code

I = 1,...,MWART

where MWART is the maximum artillery unit weapon code.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: MWART

COMMON/SIGDEF/

CONTENTS: SIGDEF contains the single shot deflection errors about the aim point for individual weapons in an artillery firing battery (see COMMON/SIGRAN/).

ARRANGEMENT: SIGDEF is a one-dimensional array.

SIGDEF(I) (real, meters)

where

SIGDEF(I) contains the errors for weapons in the I^{th} type artillery weapon code batteries

for I = 1,...,N where N is the maximum artillery weapon code.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: N

COMMON/SIGFFE/

CONTENTS: SIGFFE contains the standard deviation of the distribution of time for an artillery battery of specified weapon code to load, lay, and fire subsequent rounds of a fire-for-effect mission (see COMMON/UFFE/).

ARRANGEMENT: SIGFFE is a one-dimensional array:

SIGFFE(I) (real, seconds)

SIGFFE(I) contains the time for the I^{th} artillery fire support weapon code for
I = 1,...,MWART

where MWART is the maximum artillery unit weapon code.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: MWART

COMMON/SIGONP/

CONTENTS: SIGONP stores an array containing the standard deviations of the distribution of time required to process the original fire request for an on-call target over each radio net.

ARRANGEMENT: SIGONP is a one-dimensional array:

SIGONP(I) (real, seconds)

I = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/SIGPBR/

CONTENTS: SIGPBR stores an array containing the standard deviations of the distribution of target priorities reported by FO Bravo of type K requesting fire of type L.

ARRANGEMENT: SIGPBR(K, L) (real)

where

$$K = \begin{cases} 1 & \text{regular FO Bravo} \\ 2 & \text{special FO Bravo} \end{cases}$$

L = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 2 * MWART

COMMON/SIGPLT/

CONTENTS: SIGPLT stores an array containing the standard deviations of the distribution of time required to process an original fire request against a target of opportunity over each radio net.

ARRANGEMENT: SIGPLT is a one-dimensional array:

SIGPLT(I) (real, seconds)

I = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/SIGRAN/

CONTENTS: SIGRAN contains the single shot range errors about the aim point for individual weapons in an artillery firing battery (see COMMON/SIGDEF/).

ARRANGEMENT: SIGRAN is a one-dimensional array.

SIGRAN(I) (real, meters)

where

SIGRAN(I) contains the errors for weapons in the I^{th} type artillery weapon code batteries

for I = 1,...,N where N is the maximum artillery weapon code.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: N

COMMON/SIGRBR/

CONTENTS: SIGRBR stores an array containing the standard deviations of the distribution of the number of rounds requested by FO Bravo of type I requesting fire of type J.

ARRANGEMENT: SIGRBR(I,J) (real)

where

$$I = \begin{cases} 1 & \text{regular FO Bravo} \\ 2 & \text{special FO Bravo} \end{cases}$$

J = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 2 * MWART

COMMON/SIGSEN/

CONTENTS: SIGSEN contains the standard deviation of the time distribution whose mean is USEN.

ARRANGEMENT: SIGSEN(I)

where

I = 1,...,NTFRNT fire request nets.

INITIALIZATION: Required

USERS: MFB,AIRFB

LENGTH: NTFRNT

COMMON/SKY/

CONTENTS: SKY contains miscellaneous constants required in the aerial vehicle detection model to describe atmospheric conditions.

ARRANGEMENT: SKY is a three-dimensional array.

SILL = the sky (ambient) illuminance level (real, lumens/feet^2)

SL = the sky luminance level (real, foot-lamberts)

VR = meteorological visibility range (real, meters).

INITIALIZATION: Required

USERS: DETH

LENGTH: 3

COMMON/SLSTIM/

CONTENTS: SLSTIM contains the latest possible starting time of an artillery scheduled fire mission.

ARRANGEMENT: SLSTIM is a two-dimensional array.

SLSTIM(I,J) (real, seconds)

SLSTIM(I,J) contains the time for the I^{th} mission for the J^{th} battery for I = 1,...,6; J = 1,...,N

where N is the number of batteries.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: 6 * N

COMMON/SPDPHS/

CONTENTS: SPDPHS contains the speed at which a maneuver unit is allowed to travel after it has reached a phase line when it is waiting for other attacking maneuver units to reach the phase line.

ARRANGEMENT: SPDPHS is a two-dimensional array

where

SPDPHS(I,J) = the speed at the I^{th} phase line for the J^{th} maneuver unit

for I = 1,...,4
J = 1,...,N where N is the total number of maneuver units.

INITIALIZATION: Required for all phase line included.

USERS: MVCON

LENGTH: 4*N

COMMON/SPDSE/

CONTENTS: SPDSE contains the desired speed recorded for each aerial section while operating independently.

ARRANGEMENT: SPDSE(I) (real, meters/second)

where

I = 1,...,N aerial section numbers.

INITIALIZATION: None

USERS: SECPRM,RTATAK,ATKPRM

LENGTH: N

COMMON/SPEEDS/

CONTENTS: SPEEDS contains the speed for aerial sections with specified weapon codes performing independent movement of a given type.

ARRANGEMENT: SPEEDS(I,J) (real, meters/second)

where

$$I = \begin{cases} 1 \text{ for enroute movement} \\ 2 \text{ for loiter} \\ 3 \text{ for target attack} \\ 4 \text{ for target search, and} \end{cases}$$

J = 1,...,N aerial section weapon codes (see COMMON/ANGLIM/).

INITIALIZATION: Required

USERS: SECPRM

LENGTH: 4 * N

COMMON/SPHAS/

CONTENTS: SPHAS contains the slope of the phase lines at the points specified in common areas XPHAS and YPHAS. (Note: To guard against machine overflow, underflow or divide checks, SPHAS should be between $\pm$ .001 and $\pm$ 1000.)

ARRANGEMENT: SPHAS is a two-dimensional array

where

SPHAS(I,J) contains the slope of the I^{th} phase line for the J^{th} maneuver unit (real, meters)

where I = 1,2,...,NMPHAS (see COMMON/NPHAS/)
J = 1,2,...,N, where N is the number of maneuver units.

INITIALIZATION: Required if phase lines are used.

USERS: MVCON

LENGTH: NMPHAS*N

COMMON/SPINC/

CONTENTS: SPINC contains the time between launches desired for indirect-fire MISTIC attacks from launchers belonging to units with specified MISTIC weapon codes.

ARRANGEMENT: SPINC(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB

LENGTH: MWMIS-MWART

COMMON/SRPTS/

CONTENTS: SRPTS contains slant ranges to the target in meters.

ARRANGEMENT: SRPTS is a one-dimensional array.

SRPTS(I),

I = 1, ..., N

where SRPTS(I) is the slant range at distance point I,

SRPTS(1) = 0.0,

SRPTS(N) = the maximum slant range (meters) at which beam acquisition can occur, and

N = number of distance points.

INITIALIZATION: Required

USERS: FINAL

LENGTH: N

COMMON/SSR/

CONTENTS: SSR contains the standard deviation of time distribution whose mean is TBSR.

ARRANGEMENT: SSR(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: LAUNCH

LENGTH: MWMIS-MWART

COMMON/STNEUT/

CONTENTS: STNEUT contains the neutralization time affecting battery I when incoming projectiles are within a given distance of the battery (see COMMON/CBNEUT/).

ARRANGEMENT: STNEUT(I) contains the neutralization time for battery I (real, seconds)

for I = 1, ..., N

where

N is the total number of firing batteries.

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH

LENGTH: N

COMMON/STRMIS/

CONTENTS: STRMIS contains the time required for a firer to communicate a verification request over a given radio net.

ARRANGEMENT: STRMIS(I) (real, seconds)

where

I = 1,...,NTFRNT fire request nets.

INITIALIZATION: Required

USERS: MFB

LENGTH: NTFRNT

COMMON/STRTIM/

CONTENTS: STRTIM contains the time at which firing data will become available for each fire request on each radio net's fire request list.

ARRANGEMENT: STRTIM is a two-dimensional array:

STRTIM(I,J) (real, seconds)

I = 1,...,5 position on FRL

J = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets

INITIALIZATION: None

USERS: AFASGN, AFB, AFDC, AFO, AFSC, AIRFB, BTLETH, DATATM, FSCMON, LAUNCH, MFB, WRTFRL

LENGTH: 5 * NTFRNT

COMMON/STRTMS/

CONTENTS: STRTMS contains the time at which an artillery firing battery's present fire mission began.

ARRANGEMENT: STRTMS is a one-dimensional array.

STRTMS(I) (real, seconds)

STRTMS(I) contains the time for the I^{th} battery for I = 1,...,N

where N is the number of batteries.

INITIALIZATION: Zero

USERS: AFB

LENGTH: N

COMMON/TARASP/

CONTENTS: TARASP contains aspect angle intervals for interpolation of hit and kill probabilities.

ARRANGEMENT: TARASP contains two sets of data as follows:

1. NA - the number of values in the array ASP (integer)

2. ASP(I) where I = 1,...,NA contains aspect angle intervals such that ASP(I) < ASP(I + 1) (real, radians).

INITIALIZATION: Required.

USERS: SPDASP

LENGTH: NA + 1

COMMON/TARSPD/

CONTENTS: TARSPD contains target speed intervals for interpolation of hit and kill probabilities.

ARRANGEMENT: TARSPD contains two sets of data as follows:

1. NS - the number of values in the array SPD
2. SPD(I) where I = 1,...,NS contains speed intervals such that SPD(I) < SPD(I + 1) (real, meters/sec.)

Note: SPD(1) should be initialized to zero.

INITIALIZATION: Required.

USERS: SPDASP

LENGTH: NS + 1

COMMON/TBFR/

CONTENTS: TBFR contains the mean of the distribution of times required for a launcher belonging to a unit with a given MISTIC weapon code to set up to fire a first round.

ARRANGEMENT: TBFR(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB,LAUNCH

LENGTH: MWMIS-MWART

COMMON/TBL/

CONTENTS: TBL contains the mean of distribution of times required to complete one loading operation on a launcher belonging to a unit with a specified MISTIC weapon code.

ARRANGEMENT: TBL(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: LOADM, LAUNCH

LENGTH: MWMIS-MWART

COMMON/TBSR/

CONTENTS: TBSR contains the mean of the distribution of times required for a launcher belonging to a unit with a given MISTIC weapon code to set up to fire second and subsequent rounds.

ARRANGEMENT: TBSR(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: LAUNCH

LENGTH: MWMIS-MWART

COMMON/TC/

CONTENTS: Common TC contains the Y-coordinate of both red and blue force strongpoints.

ARRANGEMENT: TC is a one-dimensional array;

TC(I)
where
I = 1,...,NRSP subscripts for the red strongpoints

I = NRSP + 1,...,NRSP + NBSP subscripts for the blue strongpoints

where NRSP is the number of red strongpoints and NBSP is the number of blue strongpoints, user supplied storage parameters.

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: NRSP + NBSP

COMMON/TCRIT/

CONTENTS: TCRIT contains the minimum time remaining for a helicopter of specified weapon code. The values are used to determine when a helicopter is to be considered a casualty.

ARRANGEMENT: TCRIT is a one-dimensional array.

TCRIT(I) (real, seconds)

where

I = 1,...,N helicopter weapon codes (see COMMON/ANGLIM/ for a definition of weapon code).

INITIALIZATION: Required

USERS: GETHEL

LENGTH: N

COMMON/TDFRDY/

CONTENTS: TDFRDY contains the battle time at which a given launcher achieved the direct-fire activity state specified by his entry in the array LFLAG.

ARRANGEMENT: TDFRDY(I) (real, seconds)

where

I = 1,...,ITOTLN launcher numbers.

INITIALIZATION: None

USERS: MFB,LOADM,LAUNCH,ATKPRM

LENGTH: ITOTLN

COMMON/TDUD/

CONTENTS: TDUD contains the time required for a launcher belonging to a unit with a given MISTIC weapon code to react to a missile misfire.

ARRANGEMENT: TDUD(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB,LAUNCH

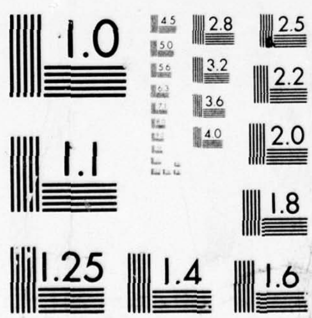
LENGTH: MWMIS-MWART

AD-A040 053 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM), VOLUME 2, SECTION--ETC(U)
DEC 71 G M CLARK, R J WILHELM DAAH01-70-C-0713

UNCLASSIFIED RF-2995-FR-71-2(U)-SECT-1 NL

3 OF 5
AD
A040053

04005



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

COMMON/TEC/

CONTENTS: Common TEC contains the effective range for a given enemy strongpoint weapon type against a dismounted CSW unit.

ARRANGEMENT: TEC is a one-dimensional array:

TEC(I) (real, meters)

where I = 1,..., Z + NRWP + NBWP

and $0 < TEC(1) \leq TEC(2)...$

NRWP, NBWP are defined in Common SPTS as

NRWP = number of red weapon types
NBWP = number of blue weapon types.

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: NRWP + NBWP + 2

COMMON/TEW/

CONTENTS: Common TEW contains the effective range for a weapon with a specified weapon code.

ARRANGEMENT: TEW is a one-dimensional array:
TEW(I)
I = 1,...,MNFRT weapon code

where MNFRT is the total number of element weapon codes.

INITIALIZATION: Required

USERS: CSRTSL

LENGTH: MNFRT

COMMON/TF/

CONTENTS: TF contains the nominal value of flight time from launch corresponding to a distance RM (given in COMMON/RM).

ARRANGEMENT: RF(I) (real, seconds)

$I = 1, \cdots, N$

where N = number of distance points and

$TF(1) = 0.0$

INITIALIZATION: Required

USERS: FLIGHT

LENGTH: N

COMMON/TFDCKL/

CONTENTS: TFDCKL contains the time at which a neutralized fire direction center will again become active, as a result of counterbattery fire.

ARRANGEMENT: TFDCKL is a one-dimensional array.

TFDCKL(I) (real, seconds)

where

$$TFDCKL(I) = \begin{cases} 0 & \text{fire direction center I is not neutralized} \\ T & \text{time at which the direction center may become active} \end{cases}$$

for $I = 1, \ldots, N$ where N is the maximum number of artillery fire direction centers.

INITIALIZATION: None

USERS: AFDC, BTLETH

LENGTH: N

COMMON/TFLY

CONTENTS: TFLY contains the computed time required to prepare a point-fire weapon for launch from a given helicopter or the computed time of flight of a beam-rider missile launched from a given helicopter.

ARRANGEMENT: TFLY(I) (real, seconds)

where

I = 1,...,N helicopter numbers.

INITIALIZATION: None

USERS: TFCOMP,ATKPRM,RTATAK

LENGTH: N

COMMON/THBM/

CONTENTS: THBM contains the mean time required to launch a direct-fire missile of a given type from a helicopter having a given aerial weapon code.

ARRANGEMENT: THBM(I,J) (real, seconds)

where

$$I = \begin{cases} 1 & \text{for a MISTIC missile} \\ 2 & \text{for a beam-rider missile, and} \end{cases}$$

J = 1,...,NA3, where NA3 is the maximum number of weapon codes assigned to aerial vehicle elements. Note that weapon codes assigned in COMMON /LWCOD/ are numbered 1,...,MAXLWC for ground elements and from MAXLWC + 1,...,MAXWC for aerial vehicle elements. Thus, NA3 = MAXWC - MAXLWC.

INITIALIZATION: Required

USERS: TFCOMP

LENGTH: 2 * NA3

COMMON/TIFRDY/

CONTENTS: TIFRDY contains the battle time at which a given launcher achieved the indirect-fire activity state specified by his entry in the array IFRFL.

ARRANGEMENT: TIFRDY(I) (real, seconds)

where

I = 1,...,ITOTLN launcher numbers.

INITIALIZATION: None

USERS: MFB, LOADM, LAUNCH

LENGTH: ITOTLN

COMMON/TIMBE/

CONTENTS: TIMBE contains the minimum time between fire missions for artillery firing batteries.

ARRANGEMENT: TIMBE is a one-dimensional array.

TIMBE(I) (real, seconds)

TIMBE(I) contains the time for the I^{th} battery for I = 1,...,N

where N is the number of batteries.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: N

COMMON/TINIT/

CONTENTS: TINIT contains times at which messages are transmitted or times at which targets are selected.

ARRANGEMENT: TINIT is a one-dimensional array (real, seconds).

$$TINIT(I) = \begin{cases} \text{time at which an FO selects a target or} \\ \quad \text{time at which an FO transmits a fire} \\ \quad \text{adjustment message if } I \le ITOTFO \\ \text{time at which an aerial unit requests} \\ \quad \text{mission confirmation if } I > ITOTFO. \end{cases}$$

INITIALIZATION: None

USERS: AIRFB,AFO,FSCMON,REVAL,AFSC

LENGTH: ITOTFO + NUMAVT

COMMON/TLOAD/

CONTENTS: TLOAD contains the computed time required to perform the next loading operation aboard each launcher (negative if not computed previously).

ARRANGEMENT: TLOAD(I) (real, seconds)

where

I = 1,...,ITOTLN launcher numbers.

INITIALIZATION: Initialized at -2.

USERS: LOADM,LAUNCH

LENGTH: ITOTLN

COMMON/TMISUN/

CONTENTS: TMISUN contains the time at which each aerial unit enters its mission operations area.

ARRANGEMENT: TMISUN(I) (real, seconds)

where

I = 1,2,...,NUMAVT aerial unit numbers.

INITIALIZATION: None

USERS: MISEND,HELFIR,CBCONT

LENGTH: NUMAVT

COMMON/TMNTD/

CONTENTS: Common TMNTD contains the time at which the dismounted crews of a specified APC will have returned from their firing position and be ready to remount.

ARRANGEMENT: TMNTD(I) (real, seconds)

I = 1,...,NAPCMX APC number

where NAPCMX is the maximum number of APC's in the battle.

INITIALIZATION: Zero, altered during execution

USERS: CSWCAS, CSWCON, MOUNT

LENGTH: NAPCMX

COMMON/TNEUTM/

CONTENTS: TNEUTM contains neutralization times for various weapon types as a result of artillery volleys or impacting missiles which did not have guidance to the target on impace.

ARRANGEMENT: TNEUTM(I,J) contains the neutralization time interval for the J^{th} weapon system type by the I^{th} artillery battery

I = 1,..., NUMART

J = 1,...,N where NUMART is the number of artillery firing batteries and N is the maximum weapon system code value specified in COMMON/LWSYS/.

INITIALIZATION: Required, not altered during execution.

USERS: ARLETH, FLIGHT

LENGTH: NUMART * N

SCENARIO DEPENDENT

Routines using Common TNEUTM may need to be recompiled to obtain the desired maximum value for subscript I.

COMMON/TPMKH/

CONTENTS: TPMKH contains kill probabilities for nontank type targets.

ARRANGEMENT: TPMKH(I,J,K) (integer)

where

I = 1,...,6	subscripts corresponding to range intervals as defined in COMMON/HPRNG/
J = 1,...,5	subscripts corresponding to kill types (e.g., 0,...,4)
K = 1,...,N	subscripts corresponding to firer-ammo-target combinations not including tank-target combinations (e.g., KCHAR-MTNKLT where KCHAR is the fourth variable in COMMON/IAMMO/ and MTNKLT is from COMMON/MAXWEP/.

INITIALIZATION: Required.

USERS: TLETH

LENGTH: $6 \cdot 5 \cdot N$

COMMON/TRET/

CONTENTS: TRET contains the time at which each helicopter element will become a casualty.

ARRANGEMENT: TRET is a one-dimensional array.

TRET(I) (real, seconds)

where

I = 1,...,N helicopter element numbers.

INITIALIZATION: Each entry should be initialized with a large value.

USERS: GETHEL,CASHEL

LENGTH: N

COMMON/TRNET/

CONTENTS: TRNET contains the battle time after which a net will be available for message transmissions.

ARRANGEMENT: TRNET is a one-dimensional array:

TRNET(I) (real, seconds)

I = 1,...,NPLT platoon net subscripts
I = NPLT + 1,...,NBTEM blue company net subscripts
I = NPLT + NBTEM + 1,...,NPLT + NTEM red company net subscripts
I = NPLT + NTEM + 1 blue battalion net subscripts
I = NPLT + NTEM + 2 red battalion net subscripts
where
NPLT = the number of platoons
NBTEM = the number of blue teams
NTEM = total number of teams

INITIALIZATION: Initialized to large negative value.

USERS: SEND, SENDC

LENGTH: NPLT + NTEM + 2

COMMON/TVMIS/

CONTENTS: TVMIS contains data used in the television guided missile model.

ARRANGEMENT: TVMIS contains ten values:

1. DT = visual fixation time for the remote flier (real, seconds)

2. ANGA = angle between horizontal and the rear limit of the camera's field of view (real, radians)

3. ANGB = angle between horizontal and the front limit of the camera's field of view (real, radians)

4. GAMM = angle between vertical and the side limits of the camera's field of view (real, radians)

5. SDANG = standard deviation of flier's estimate of target heading (real, radians)

6. CSDV = constant associated with standard deviation of flier's estimate of target velocity (real)

7. LAQ = flag to indicate whether target acquisition by the flier has occurred (integer)

$$\text{LAQ} = \begin{cases} 1 \sim \text{target has been acquired by flier} \\ 0 \sim \text{otherwise} \end{cases}$$

8. LDET = flag to indicate whether target was seen by the flier during his last fixation (integer)

$$\text{LDET} = \begin{cases} 1 \sim \text{target was seen during last fixation} \\ 0 \sim \text{otherwise} \end{cases}$$

9. TLOS = cumulative time during which the target has been seen by the flier (real, seconds)

10. TLOST = cumulative time since target was last seen by the flier (real, seconds)

INITIALIZATION: DT, ANGA, ANGB, GAMM, SDANG, and CSDV must be supplied as input. No initialization required for others.

USERS: FINALE, POSEST, TVDET

LENGTH: 10

COMMON/TVOPT/

CONTENTS: TVOPT contains data concerning the tactic employed for the E-O missile system in the post lock-on mode.

ARRANGEMENT: Single value.

LTVOPT = flag indicating post lock-on mode for E-O missile

$$LTVOPT = \begin{cases} 1 \text{ for human supervision} \\ 0 \text{ for electronic guidance} \end{cases}$$

INITIALIZATION: Required

USERS: TVDET, FINALE

LENGTH: 1

COMMON/TYPMIS/

CONTENTS: TYPMIS contains a flag to designate the type of flight path and search pattern of a missile from a specified launcher.

ARRANGEMENT: TYPMIS is a one-dimensional array: (integer)

$$TYPMIS(I) = \begin{cases} K < 3 \text{ if a level flight missile} \\ K \geq 3 \text{ if ballistic flight missile} \\ K \text{ which is odd if a MISTIC missile} \\ K \text{ which is even if an E-O missile} \\ 2 < K < 5 \text{ if seeker is fixed to the missile} \\ K > 4 \text{ if seeker is stabilized to launch angle orientation} \end{cases}$$

I = 1,...,ITOTLN, where ITOTLN is the total number of launchers, a user-supplied storage dimension parameter.

INITIALIZATION: Required

USERS: BFLITE, FLIGHT, LAUNCH, SEEKER

LENGTH: ITOTLN

COMMON/UADJ/

CONTENTS: UADJ contains the mean time required to adjust firing data after receipt of an adjustment message over each radio net (see COMMON/SIGADJ/).

ARRANGEMENT: UADJ is a one-dimensional array:

UADJ(I) (real, seconds)
I = 1,...,NTFRNT fire request net
where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/UAL/

CONTENTS: UAL contains the mean time for an artillery firing battery of specified weapon code to load, lay, and fire the initial volley of a fire mission (see COMMON/SIGAL/).

ARRANGEMENT: UAL is a one-dimensional array:

UAL(I) (real, seconds)

UAL(I) contains the time for the I^{th} fire support weapon code

I = 1,...,MWART

where MWART is the maximum artillery unit weapon code.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: MWART

COMMON/UFFE/

CONTENTS: UFFE contains the mean time for an artillery battery of specified weapon code to load, lay, and fire subsequent rounds of a fire-for-effect mission (see COMMON /SIGFFE/.)

ARRANGEMENT: UFFE is a one-dimensional array:

UFFE(I) (real, seconds)

UFFE(I) contains the time for the I^{th} fire support weapon code

I = 1,...,MWART

where MWART is the maximum artillery unit weapon code.

INITIALIZATION: Required, not altered during execution.

USERS: AFB

LENGTH: MWART

COMMON/UONP/

CONTENTS: UONP contains the mean time required to process the original fire request for an on-call target over each radio net (see COMMON/SIGONP/).

ARRANGEMENT: UONP is a one-dimensional array:

UONP(I) (real, seconds)

I = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/UPLT/

CONTENTS: UPLT contains the mean time required to process an original fire request against a target of oppurtunity over each radio net.

ARRANGEMENT: UPLT is a one-dimensional array:

UPLT(I) (real, seconds)

I = 1,...,NTFRNT fire request net

where NTFRNT is the total number of fire request nets.

INITIALIZATION: Required

USERS: AFDC

LENGTH: NTFRNT

COMMON/UPRBRA/

CONTENTS: UPRBRA stores an array containing the means of the priority distribution whose standard deviations are stored in COMMON/SIGPBR/.

ARRANGEMENT: UPRBRA(I,J) (real)

where

$$I = \begin{cases} 1 & \text{regular FO Bravo} \\ 2 & \text{special FO Bravo} \end{cases}$$

J = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 2 * MWART

COMMON/URNBRA/

CONTENTS: URNBRA stores an array containing the means of the distributions of rounds requested, whose standard deviations are stored in COMMON/SIGRBR/.

ARRANGEMENT: URNBRA(I,J) (real)

where

$$I = \begin{cases} 1 & \text{regular FO Bravo} \\ 2 & \text{special FO Bravo} \end{cases}$$

J = 1,...,MWART fire support weapon code.

INITIALIZATION: Required

USERS: AFO

LENGTH: 2 * MWART

COMMON/USEN/

CONTENTS: USEN contains the mean of the distribution of times required by an FO communicating over radio net I to verify a target for a firer.

ARRANGEMENT: USEN(I) (real, seconds)

where

I = 1,...,NTFRNT.

INITIALIZATION: Required

USERS: MFB,AIRFB

LENGTH: NTFRNT

COMMON/VEHSPD/

CONTENTS: VEHSPD contains firer speed intervals for interpolation of hit and kill probabilities.

ARRANGEMENT: VEHSPD contains two sets of data as follows:

1. NV - the number of values in the array SPD
2. SPD(I) where I = 1,...,NS contains speed intervals such that SPD(I) < SPD(I+1)

   Note: SPD(1) should be initialized to zero.

INITIALIZATION: Required.

USERS: SPDASP

LENGTH: NV + 1

COMMON/VM/

CONTENTS: VM contains the velocity of a missile in meters/second (average velocity if level flight, initial velocity if ballistic flight).

ARRANGEMENT: VM is a one-dimensional array:

VM (ILNNUM) ILNNUM = 1,...,ITOTLN

where ITOTLN = total number of launchers.

INITIALIZATION: Required

USERS: BFLITE, FINAL

LENGTH: ITOTLN

COMMON/VULRAD/

CONTENTS: VULRAD contains an array specifying the vulnerability radius of a given weapon type to a single artillery round from a specified firing battery.

ARRANGEMENT: VULRAD is a two-dimensional array.

VULRAD(I,J) (real, meters)

where

I = 1,...,8 artillery battery number, and
J = 1,...,N target element weapon code.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: 8 * N

COMMON/WAITAD/

CONTENTS: WAITAD contains the maximum time that a fire-support element with a given weapon code will wait for mission confirmation or adjustment.

ARRANGEMENT: WAITAD(I) (real, seconds)

where

I = 1,...,MWAIR fire support weapon codes.

INITIALIZATION: Required

USERS: AIRFB,MFB

LENGTH: MWAIR

COMMON/WFUEL/

CONTENTS: WFUEL contains the fuel load remaining for each element in a given aerial vehicle section.

ARRANGEMENT: WFUEL(I) (real, pounds)

where

I = 1,...,N aerial section numbers.

INITIALIZATION: Required

USERS: SECSET,HELMOV,FUELD,RETIRE

LENGTH: N

COMMON/WPAT/

CONTENTS: WPAT stores an array containing the priority weight factor for fire pattern J used by artillery weapon code K.

ARRANGEMENT: WPAT(J,K) (real)

where

J = 1,...,5 fire pattern number
K = 1,...,MWART artillery weapon code

INITIALIZATION: Required

USERS: SELECA

LENGTH: 5 * MWART

COMMON/WPHAS/

CONTENTS: WPHAS contains the width of a phase line zone, within which a maneuver unit is considered as crossing the phase line (for a west to east attack, WPHAS is positive and for an east to west attack, WPHAS is negative).

ARRANGEMENT: WPHAS is a two-dimensional array

where

WPHAS(I,J) = the width of the I^{th} phase line for the J^{th} maneuver unit (real, meters)

for I = 1,...,4
J = 1,...,N where N is the total number of maneuver units.

INITIALIZATION: Required for all phase lines included.

USERS: MVCON

LENGTH: 4*N

COMMON/WT/

CONTENTS: WT contains the time required to prepare a MISTIC missile for launch from a launcher belonging to a unit with a given MISTIC weapon code.

ARRANGEMENT: WT(I) (real, seconds)

where

I = 1,...,MWMIS-MWART MISTIC unit weapon codes.

INITIALIZATION: Required

USERS: MFB, LAUNCH

LENGTH: MWMIS-MWART

COMMON/XCONC/

CONTENTS: XCONC stores an array containing the X-coordinates of the center of each concentration area.

ARRANGEMENT: XCONC(I) (real, meters)

where

I = 1,...,20 concentration number.

INITIALIZATION: Required

USERS: AFO, FSCHEK

LENGTH: 20

COMMON/XCSWIN/

CONTENTS: Common XCSWIN contains the value of the grade angle for the K^{th} point from which the corresponding limiting speed for dismounted CSW units is computed.

ARRANGEMENT: XCSWIN is a one-dimensional array:

XCSWIN(I)

I = 1,...,INCMAX, where INCMAX is the number of points used.

INITIALIZATION: Required

USERS: CSTIME

LENGTH: INCMAX

COMMON/XDF/

CONTENTS: Common XDF contains the X-coordinate of the primary desired firing position for a CSW unit.

ARRANGEMENT: XDF is a one-dimensional array:

XDF(I)

I = 1,...,NCSWMX CSW unit

where NCSWMX is the maximum number of CSW units in the battle.

INITIALIZATION: None

USERS: CSWCON, CSWGRD

LENGTH: NCSWMX

COMMON/XDFA/

CONTENTS: Common XDFA contains the X-coordinate of the alternate desired firing position for a CSW unit.

ARRANGEMENT: XDFA is a one-dimensional array:

XDFA(I)

I = 1,...,NCSWMX CSW unit

where NCSWMX is the maximum number of CSW units in the battle.

INITIALIZATION: None

USERS: CSWCON, CSWGRD

LENGTH: NCSWMX

COMMON/XFAC/

CONTENTS: XFAC contains the X dimension of the proportionality constant used in computing individual weapon aim points for an artillery firing battery firing a given firing pattern.

ARRANGEMENT: XFAC is a two-dimensional array.

XFAC(I,J) (real)

where

XFAC(I,J) contains the constant for the I^{th} pattern of the J^{th} artillery weapon code

for I = 1,...,5
J = 1,...,N where N is the maximum artillery weapon code.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: 5*N

COMMON/XFB/

CONTENTS: XFB stores an array containing the centroid X-coordinate of each artillery firing battery and MISTIC launcher unit.

ARRANGEMENT: XFB(I) (real, meters)

where

$$I = \underbrace{1,\ldots,\mathrm{NUMART}}_{\text{artillery firing batteries}},\underbrace{\mathrm{NUMART}+1,\ldots,\mathrm{NUMART}+\mathrm{MISTUN}}_{\text{MISTIC launcher units}}$$

INITIALIZATION: Required

USERS: AVAIL,SELECA,FSCHEK,AFSC,CBCONT

LENGTH: NUMART + MISTUN

COMMON/XLOC/

CONTENTS: XLOC contains the X dimension of the offset of each weapon in an artillery firing battery from the center of the battery.

ARRANGEMENT: XLOC is a two-dimensional array.

XLOC(I,J) (real, meters)

where

XLOC(I,J) contains the location of the I^{th} weapon for the J^{th} artillery weapon code

for I = 1,...,6
J = 1,...,N where N is the maximum number of artillery weapon codes.

INITIALIZATION: Required

USERS: ARLETH

LENGTH: 6*N

COMMON/XLSTAS/

CONTENTS: XLSTAS contains the length of the side of a loiter station of a specified type occupied by an aerial section with a given aerial weapon code.

ARRANGEMENT: XLSTAS(I,J) (real, meters)

where

I =
1 indicates section is waiting for fire support
2 indicates section is waiting at retirement position
3 indicates section is waiting at defensive position, and

J = 1,...,N aerial weapon codes (see COMMON/ANGLIM/).

INITIALIZATION: Required

USERS: RTLOIT

LENGTH: 3 * N

COMMON/XLSTAU/

CONTENTS: XLSTAU contains the length of the side of a loiter station of a given type occupied by an aerial unit with a specified weapon code.

ARRANGEMENT: XLSTAU(I,J) (real, meters)

where

$$I = \begin{cases} 1 \text{ indicates unit is awaiting a mission} \\ 2 \text{ indicates unit is waiting at retirement position} \\ 3 \text{ indicates unit is waiting at defensive position} \\ 4 \text{ indicates unit is loitering as an indirect-fire support unit, and} \end{cases}$$

J = 1,...,MWAIR-MWMIS aerial unit weapon codes.

INITIALIZATION: Required

USERS: RTLOIT

LENGTH: 4 * (MWAIR-MWMIS)

COMMON/XPHAS/

CONTENTS: XPHAS contains the X coordinate of a point on the phase line which is near the intersection of the phase line and each axis of advance.

ARRANGEMENT: XPHAS is a two-dimensional array:

XPHAS(I,J) contains the coordinate of the I^{th} phase line for the J^{th} maneuver unit (real, meters)

where
I = 1,...,NMPHAS total number of phase lines represented
J = 1,...,MNMNU number of maneuver units

INITIALIZATION: Required if phase lines are used

USERS: MVCON

LENGTH: NMPHAS * MNMNU

COMMON/XS/

CONTENTS: XS contains X coordinates of the current objective of each aerial section.

ARRANGEMENT: XS is a one-dimensional array.

XS(I) (real, meters)

where

I = 1,...,N and N is the maximum number of aerial sections.

INITIALIZATION: None

USERS: RTCROS,HELCON

LENGTH: N

COMMON/XSAVE/

CONTENTS: XSAVE contains the X coordinates computed for an aerial section attack route.

ARRANGEMENT: XSAVE(I,J) (real, meters)

where

I = 1,...,8 attack route points, and

J = 1,...,N aerial section numbers.

INITIALIZATION: None

USERS: RTATAK

LENGTH: 8 * N

COMMON/XTRIG/

CONTENTS: XTRIG stores an array containing the X-coordinate of the center of each trigger area.

ARRANGEMENT: XTRIG(I) (real, meters)

where

I = 1,...,30 trigger area number.

INITIALIZATION: Required

USERS: TRIG

LENGTH: 30

COMMON/YCSWIN/

CONTENTS: Common YCSWIN contains the value of the limiting speed for a CSW unit for a given weapon code and grade angle.

ARRANGEMENT: YCSWIN(I,J)

I = 1,...,INCMAX
J = 1,...,NCWMAX CSW weapon code

where INCMAX is the number of points used in Common XCSWIN and NCWMAX is the maximum number of weapon codes for crew served weapons.

INITIALIZATION: Required

USERS: CSTIME

LENGTH: INCMAX * NCWMAX

SCENARIO DEPENDENT

Routines using Common YCSWIN may need to be recompiled to obtain the desired maximum value for subscript I.

COMMON/YDF/

CONTENTS:	Common YDF contains the Y-coordinate of the primary desired firing position for a CSW unit.
ARRANGEMENT:	YDF is a one-dimensional array: YDF(I) I = 1,...,NCSWMX CSW unit where NCSWMX is the maximum number of CSW units in the battle.
INITIALIZATION:	None
USERS:	CSWCON, CSWGRD
LENGTH:	NCSWMX

COMMON/YDFA/

CONTENTS:	Common YDFA contains the Y-coordinate of the alternate desired firing position for a CSW unit.
ARRANGEMENT:	YDFA is a one-dimensional array: YDFA(I) I = 1,...,NCSWMX CSW unit where NCSWMX is the maximum number of CSW units in the battle.
INITIALIZATION:	None
USERS:	CSWCON, CSWGRD
LENGTH:	NCSWMX

COMMON/YFAC/

CONTENTS: YFAC contains the Y dimension of the proportionality constant used in computing individual weapon aim points for an artillery firing battery firing a given firing pattern.

ARRANGEMENT: YFAC is a two-dimensional array:

YFAC (I,J) (real)
contains
the constant for the I^{th} pattern of the J^{th} artillery weapon code
for
I = 1,...,5
J = 1,...,N where N is the maximum artillery weapon code

INITIALIZATION: Required

USERS: ARLETH

LENGTH: 5 * N

COMMON/YFB/

CONTENTS: Y analogue of XFB

COMMON/YLOC/

CONTENTS: YLOC contains the Y-dimension of the offset of each weapon in an artillery firing battery from the center of the battery.

ARRANGEMENT: YLOC is a two-dimensional array:

YLOC(I,J) (real, meters) contains the location of the I^{th} weapon for the J^{th} artillery weapon code

for I = 1,...,6
J = 1,...,N
where N is the maximum number of artillery weapon codes.

INITIALIZATION: Required

USERS: ARLETH, BTLETH

LENGTH: 6 * N

COMMON/YPHAS/

CONTENTS: YPHAS contains the Y-coordinate of a point on the phase line which is near the intersection of the phase line and each axis of advance.

ARRANGEMENT: YPHAS is a two-dimensional array:
where
YPHAS(I,J) contains the coordinate of the I^{th} phase line for the J^{th} maneuver unit (real, meters)
where
I = 1,...,NMPHAS, total number of phase lines represented
J = 1,...,MNMNU, number of maneuver units.

INITIALIZATION: Required if phase lines are used

USERS: MNCON

LENGTH: NMPHAS * MNMNU

COMMON/YSAVE/

CONTENTS: YSAVE is the Y analog of XSAVE.

COMMON/ZM/

CONTENTS: ZM contains the nominal value of missile altitude at a distance RM in meters (given in COMMON/RM)

ARRANGEMENT: ZM(I) (real, meters)

I = 1, ···, N

where N = number of distance points and

ZM(1) = 0.0

INITIALIZATION: Required

USERS: FLIGHT

LENGTH: N

COMMON/ZMD/

CONTENTS: ZMD contains the standard deviation in altitude at a distance RM (given in COMMON/RM)

ARRANGEMENT: ZMD(I) (real, meters^2)

$I = 1, \cdots, N$

where N = number of distance points and

ZMD(1) = 0.0

INITIALIZATION: Required

USERS: FLIGHT

LENGTH: N

COMMON/ZRT/

CONTENTS: ZRT contains the Z-coordinates of the movement control points for the lead element of each aerial unit measured relative to the zero elevation plane.

ARRANGEMENT: ZRT is a two-dimensional array:

ZRT(I,J) (real, meters)

where

I = 1,...,20 indices for the points in the route (see COMMON/XRT/ for arrangement), and

J = 1,...,M maneuver unit numbers.

INITIALIZATION: None

USERS: HXYMCP, RTJOIN, RTSECT

LENGTH: 20 * M

COMMON/ZSAVE/

CONTENTS: ZSAVE(I, J) is the Z analog of XSAVE.

APPENDIX B

SUBROUTINE FLOW CHARTS

List of Subroutines in Section 1

See Section 2 for remainder of subroutines.

Subroutine ADJPOS

PURPOSE: Subroutine ADJPOS determines the adjusted position of a platoon in a team or the adjusted position of a section in a platoon. The position is adjusted to account for platoons or sections not operating in the unit formation.

CALLING SEQUENCE:

```
CALL ADJPOS(NPAR,ICNT,JKPOS)
```

where

JKPOS	=	position of unit in parent organization
NPAR	=	parent organization number
ICNT	=	program process controller 1 parent organization is a team 0 parent organization is a platoon

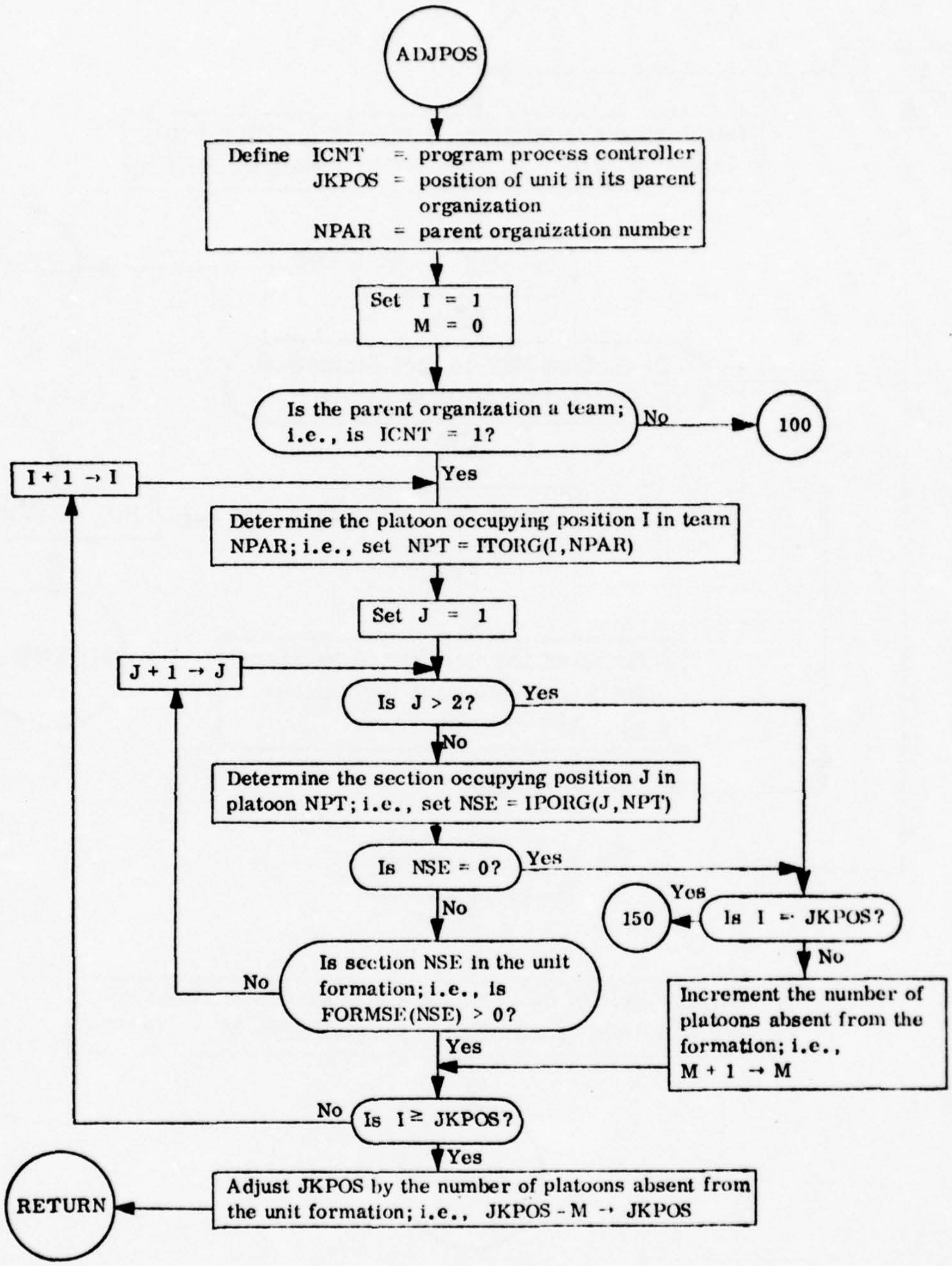
METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

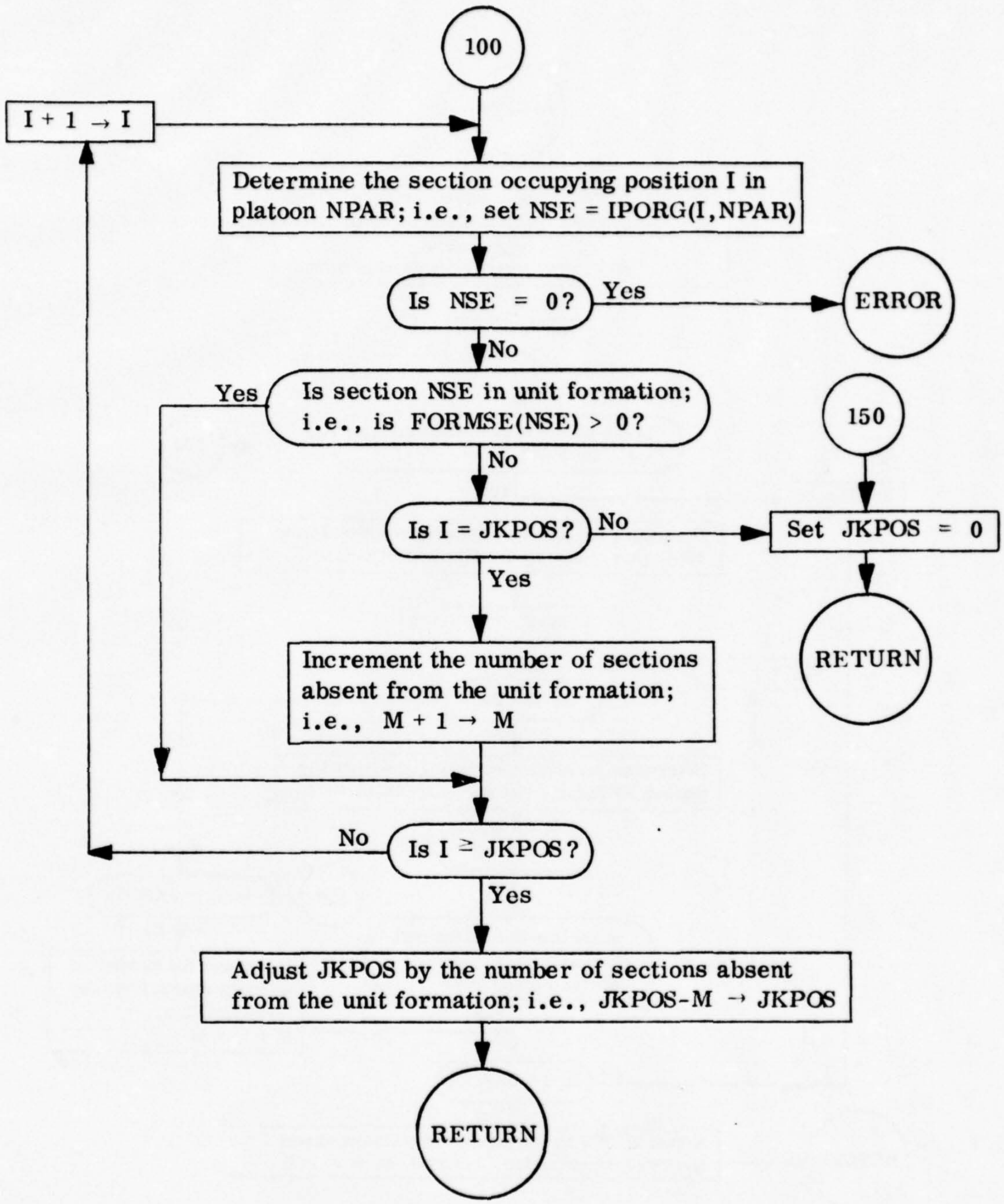
ITORG
IPORG
FORMSE

SUBROUTINES REQUIRED:

ERROR



Subroutine ADJPOS: Formation Position Adjustment



Subroutine ADJPOS: Continued

Subroutine AFASGN

PURPOSE: Subroutine AFASGN is used to determine whether aerial units should perform MISTIC forward observer missions, special forward observer missions or indirect-fire MISTIC missions.

CALLING SEQUENCE:

CALL AFASGN(KOLOR,IASGN)

where

KOLOR = indicates which force is being considered

IASGN = unit to which aerial unit is to be assigned

If IASGN < 0, an aerial unit should perform an indirect-fire MISTIC mission as a member of MISTIC unit IASGN

If IASGN > NUMART, an aerial unit should perform a MISTIC forward observer mission as a member of MISTIC unit IASGN - NUMART

If 0 < IASGN ≤ NUMART, an aerial unit should perform a special forward observer mission

If IASGN = 0, no mission should be performed

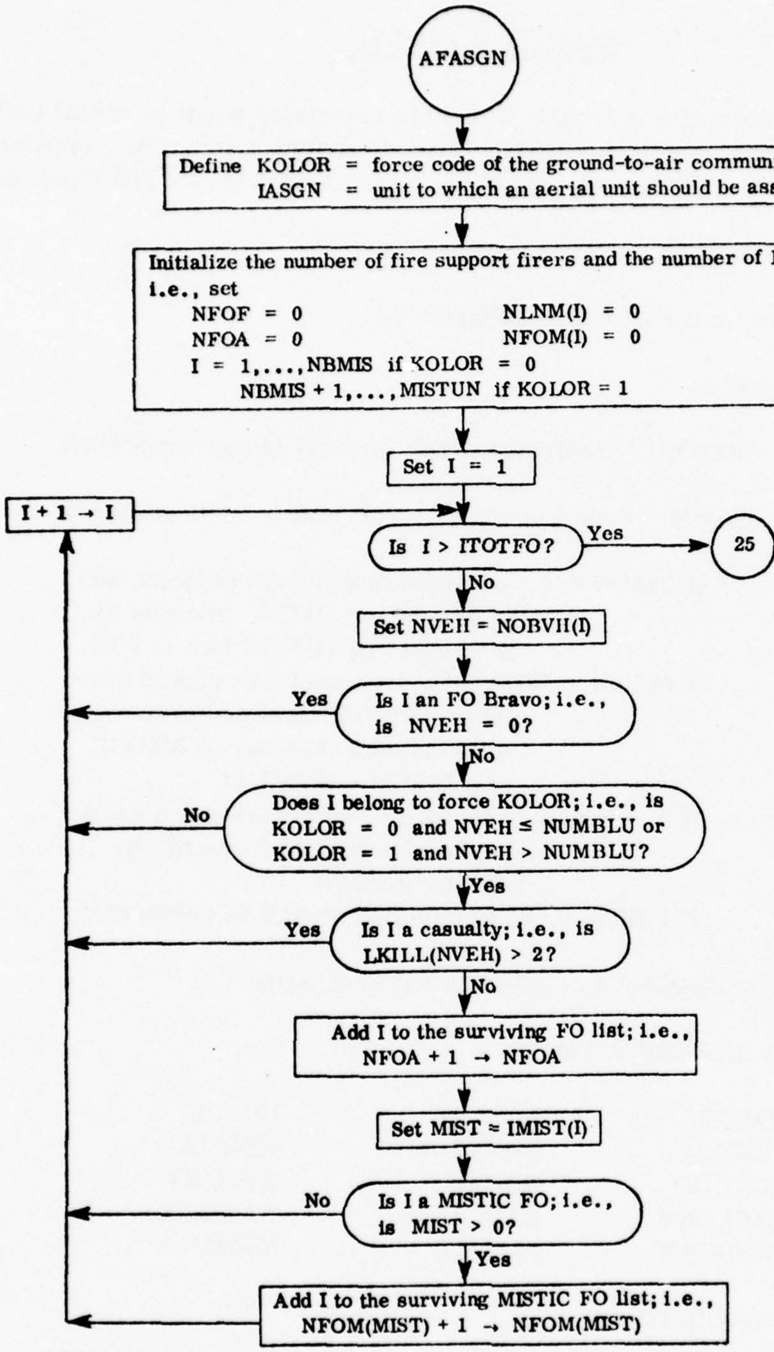
METHOD: See Chapter 2 of RF 2978 FR 71-3A (U).

COMMON AREAS REFERENCED:

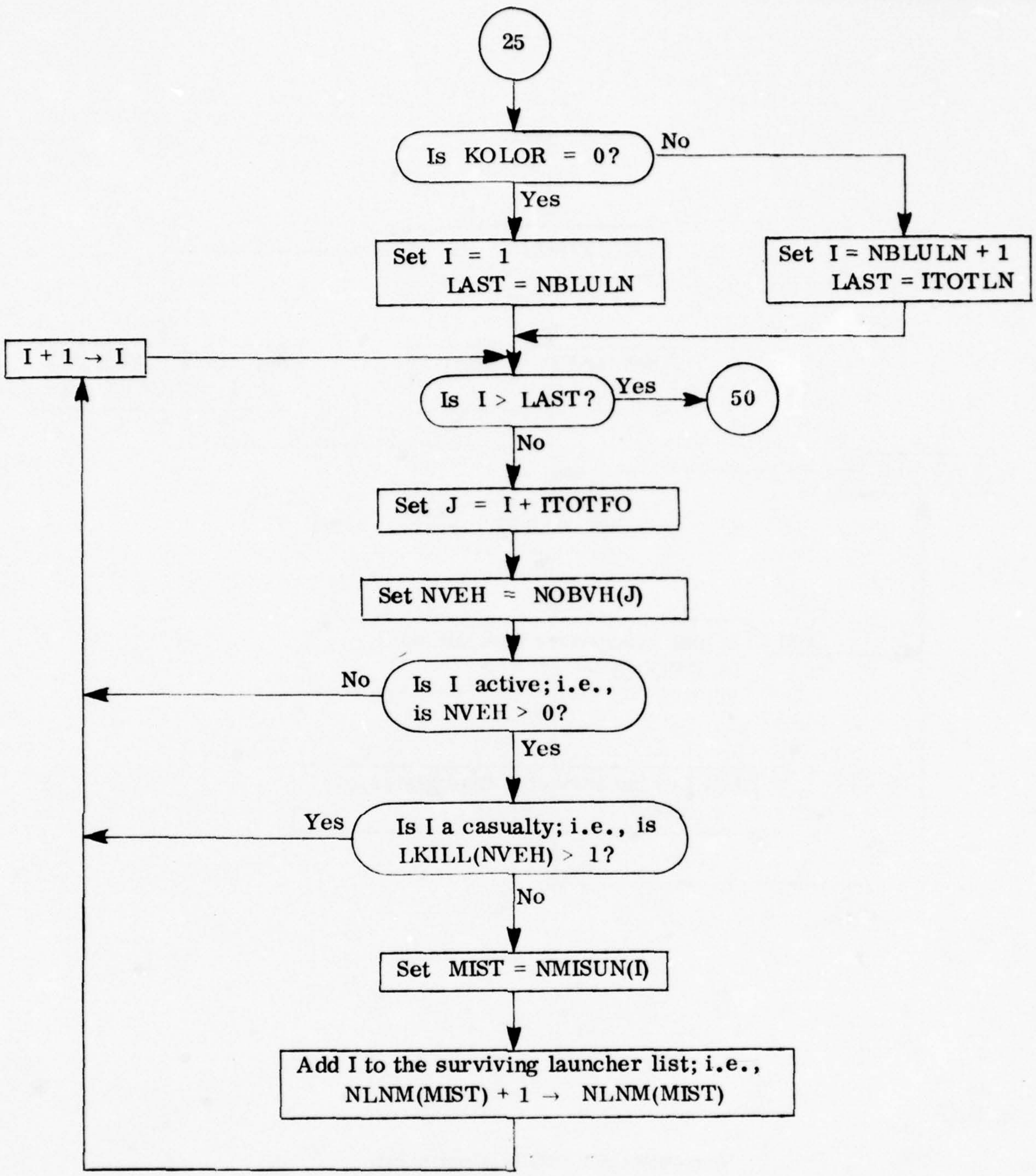
IMIST	NMISUN	IPHASE
LKILL	NUMBER	NFOMAX
LWCOD	NOBVH	NFOCRT
NFLCRT	KFDC	NLNMAX
NFUCRT	IUNACT	NFMMAX

SUBROUTINES REQUIRED:

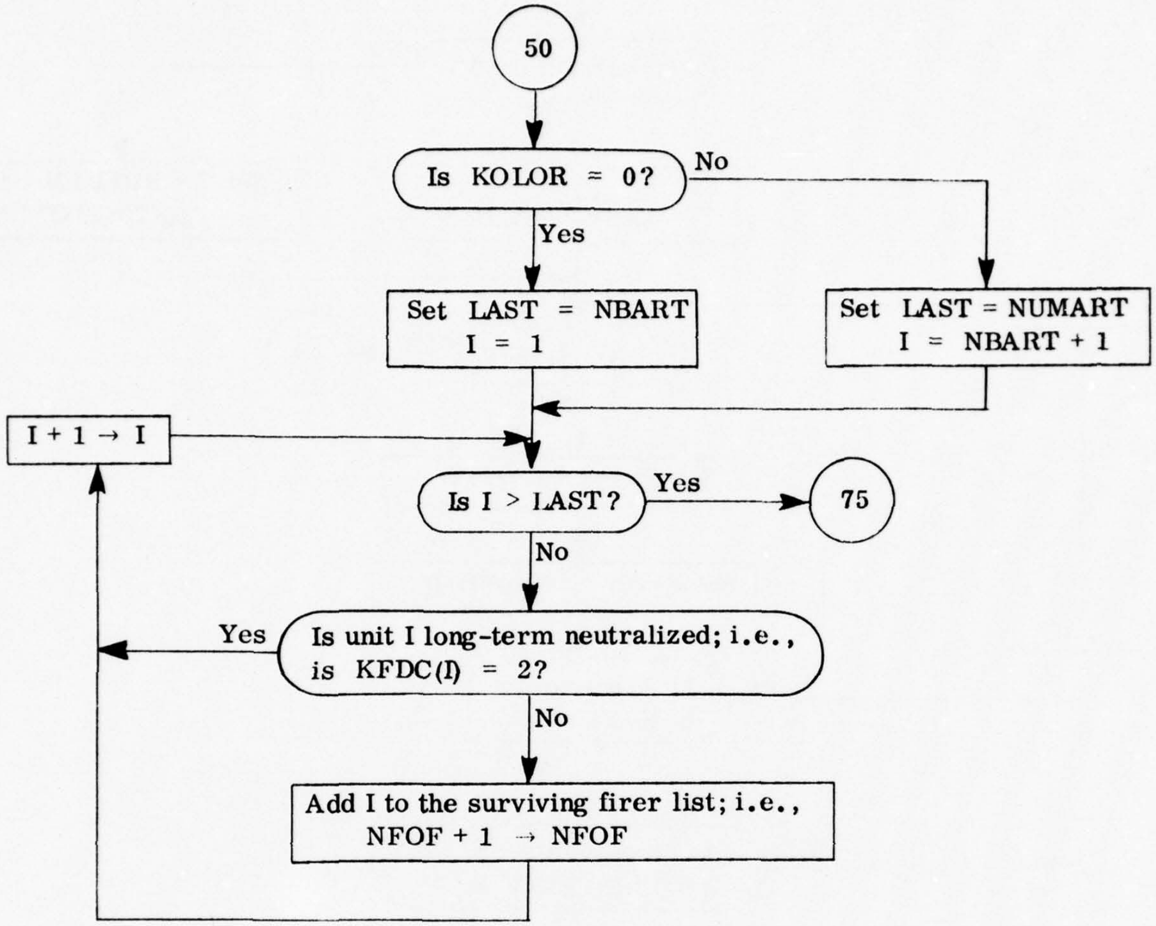
None



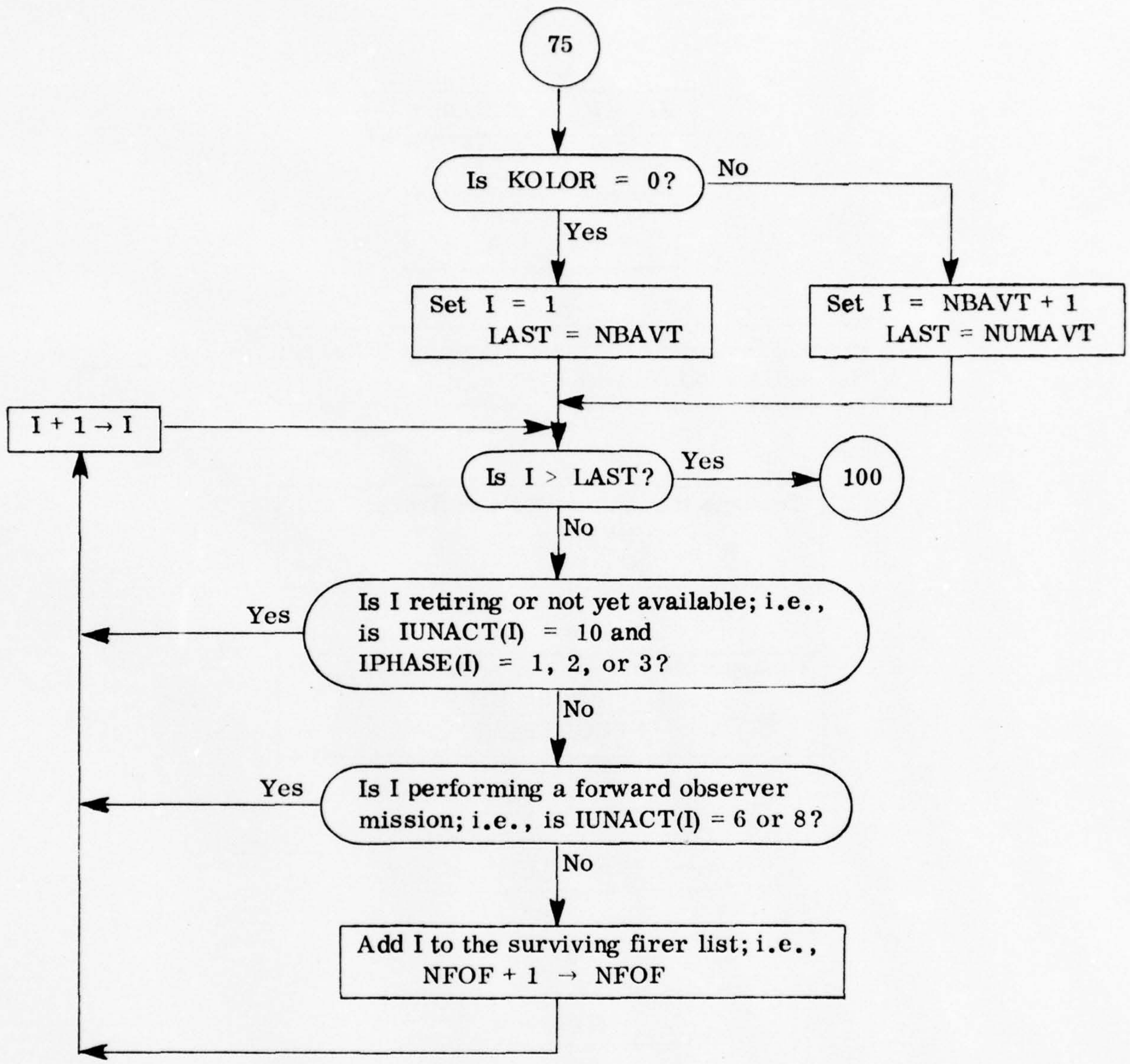
Subroutine AFASGN: Determining Requirements for Aerial Unit Launcher and Forward Observer Missions



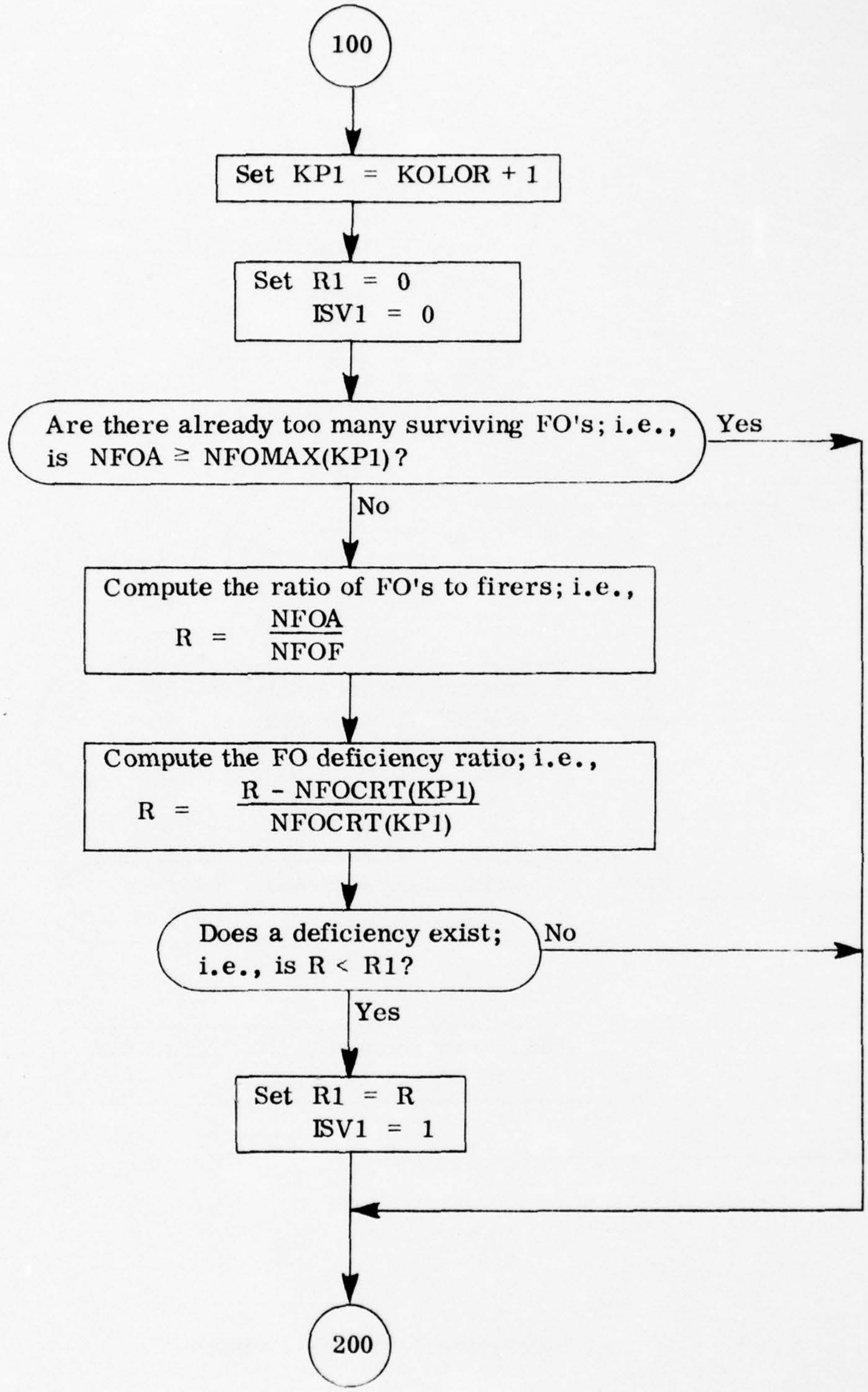
Subroutine AFASGN: Continued



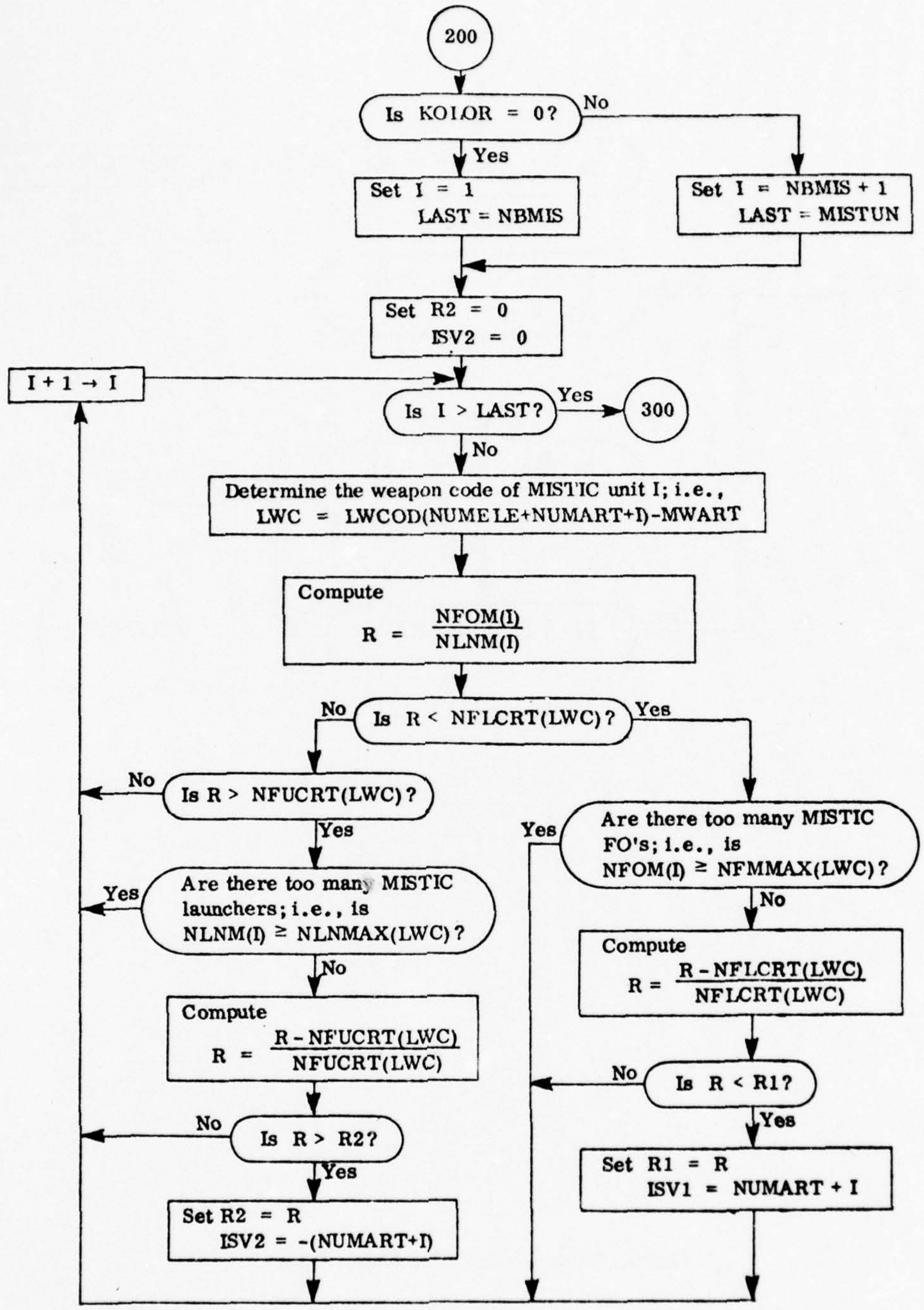
Subroutine AFASGN: Continued



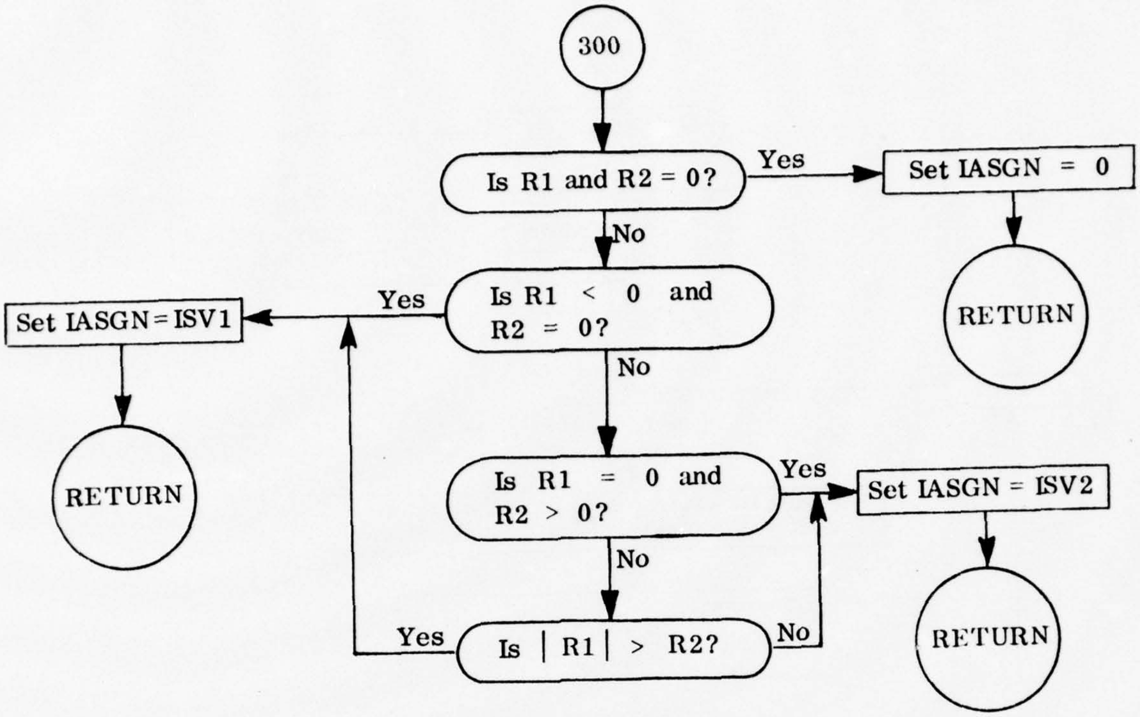
Subroutine AFASGN: Continued



Subroutine AFASGN: Continued



Subroutine AFASGN: Continued



Subroutine AFASGN: Continued

Subroutine AFDC

PURPOSE: Subroutine AFDC represents communications activities associated with initiation, confirmation and cancellation of fire requests.

CALLING SEQUENCE:

CALL AFDC(TIME,NT)

where

TIME = computed event time for radio net NT

NT = radio net number being processed

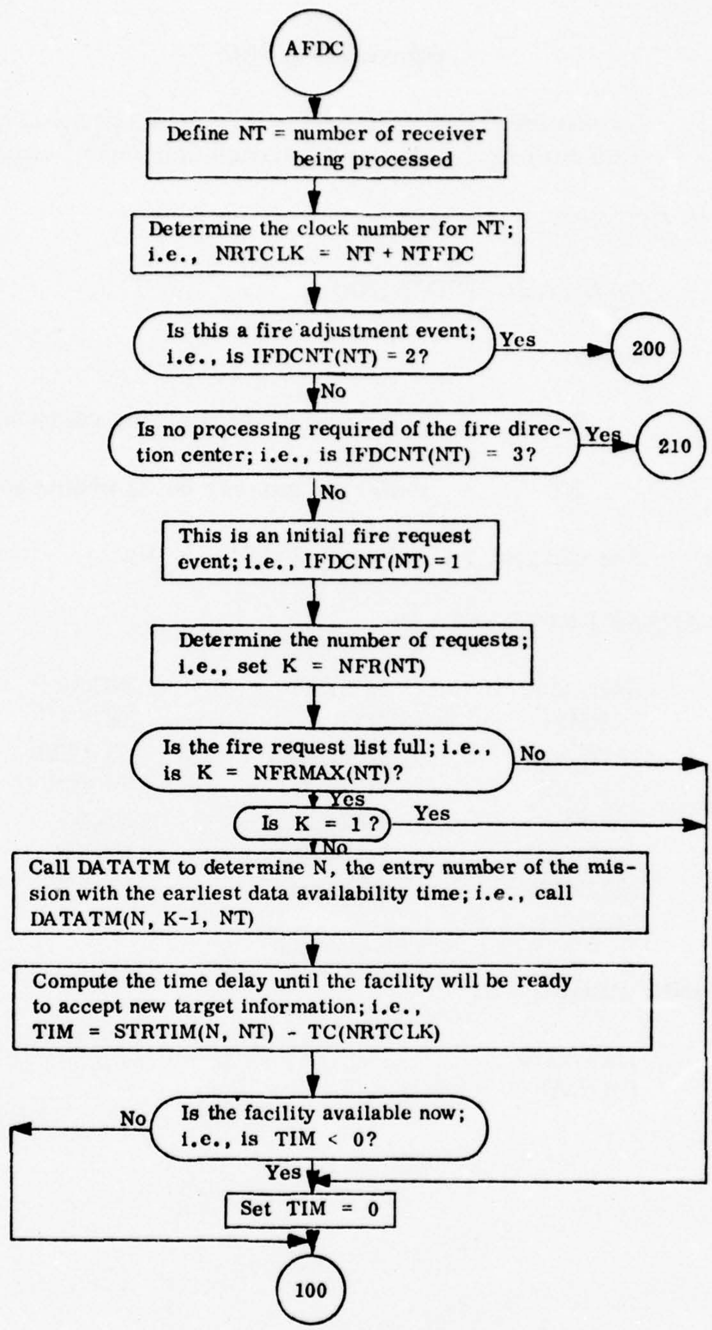
METHOD: See Chapter 3 of RF 2978 FR 71-3A (U).

COMMON AREAS REFERENCED:

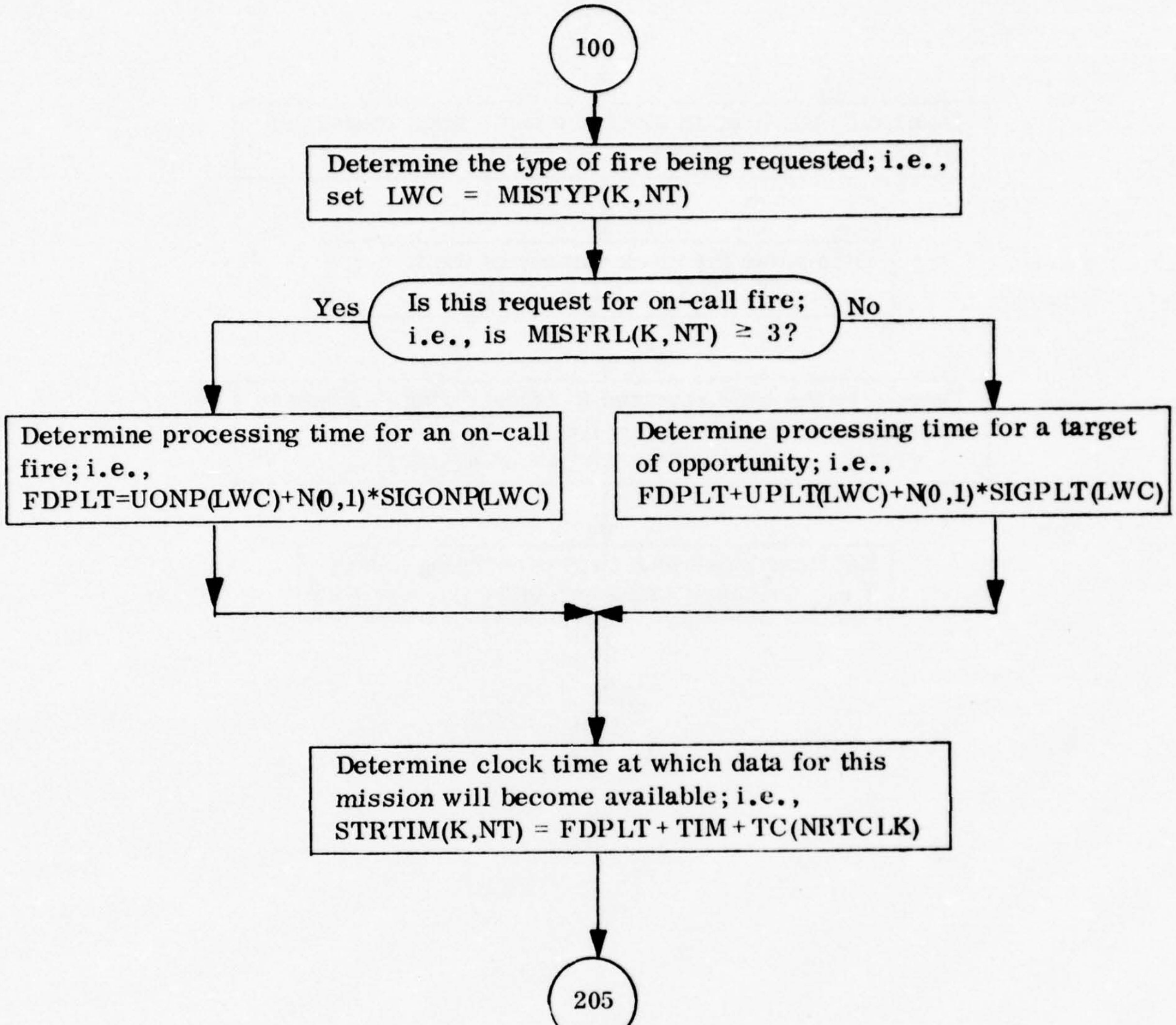
ECLOCK	MISTYP	SIGONP
IFBMIS	NFR	SIGPLT
IFDCKL	NFRMAX	STRTIM
IFDCNT	NTELE	TFDCKL
KFDC	NUMBER	UADJ
LSTFDC	NUMFIR	UONP
MAINPR	SIGADJ	UPLT
MISFRL		

SUBROUTINES REQUIRED:

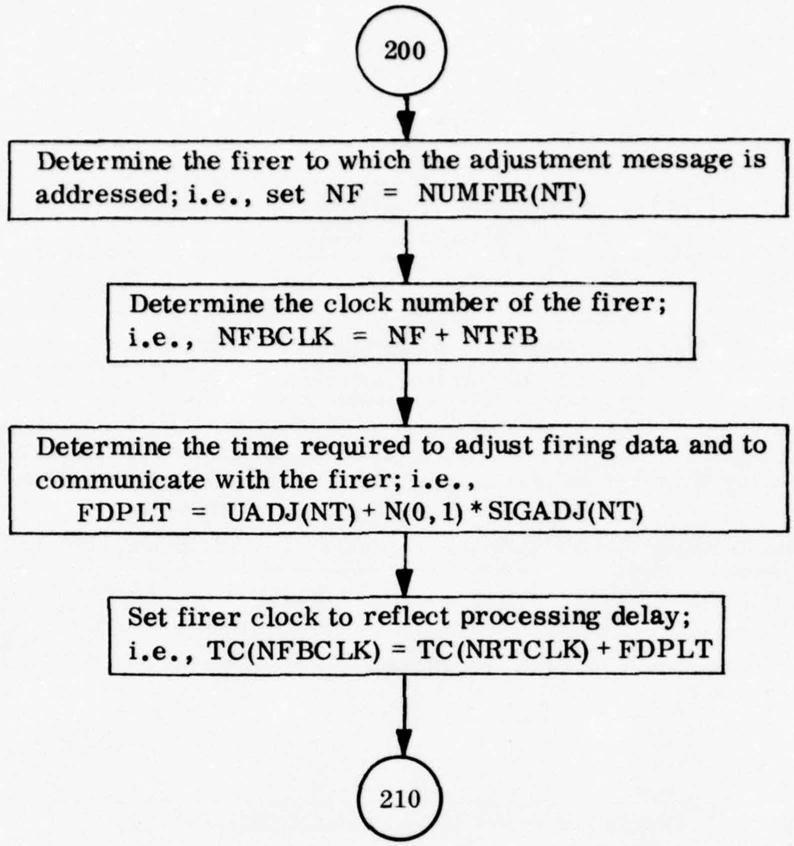
DATATM
RANND



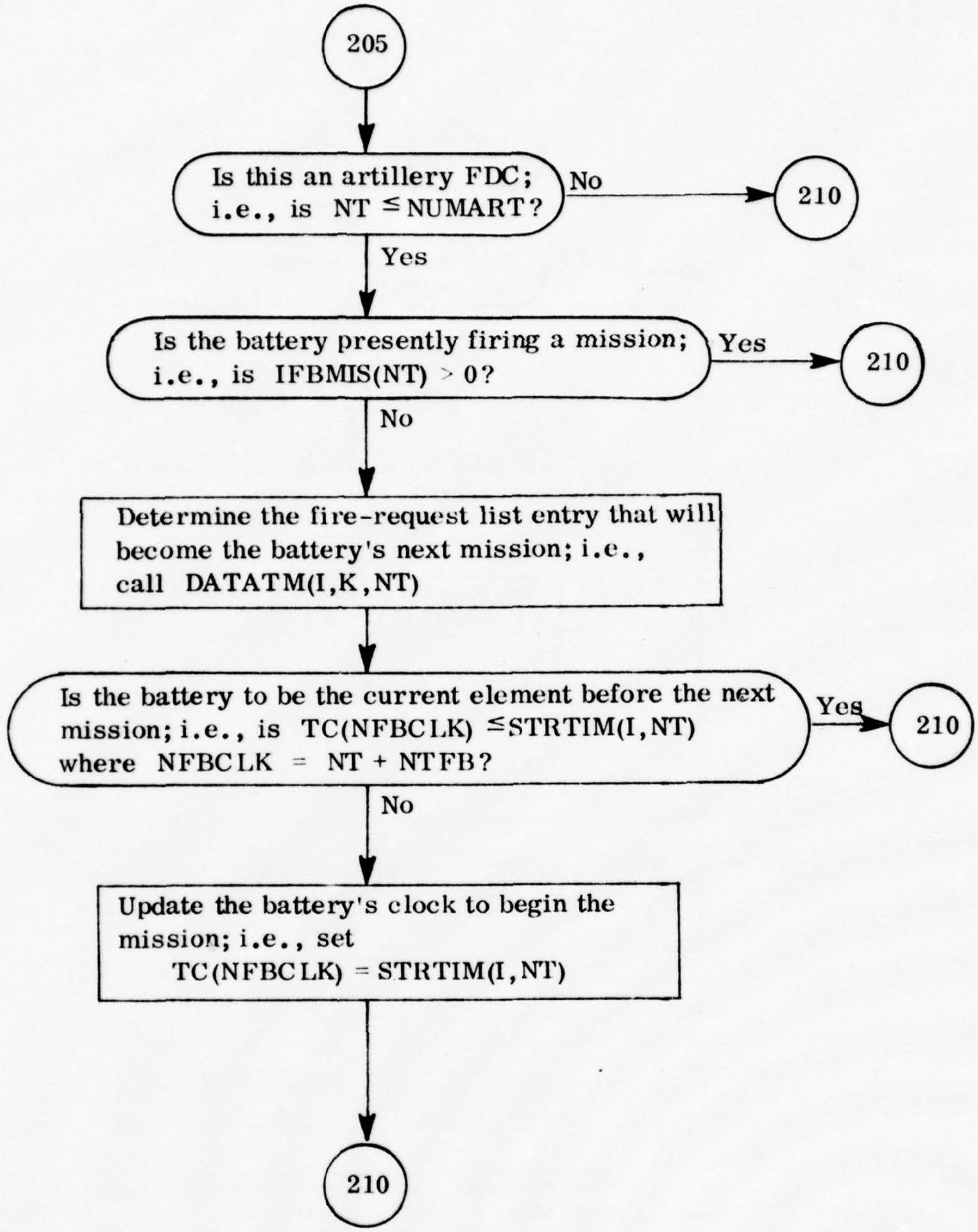
Subroutine AFDC: Fire Direction Center Activities



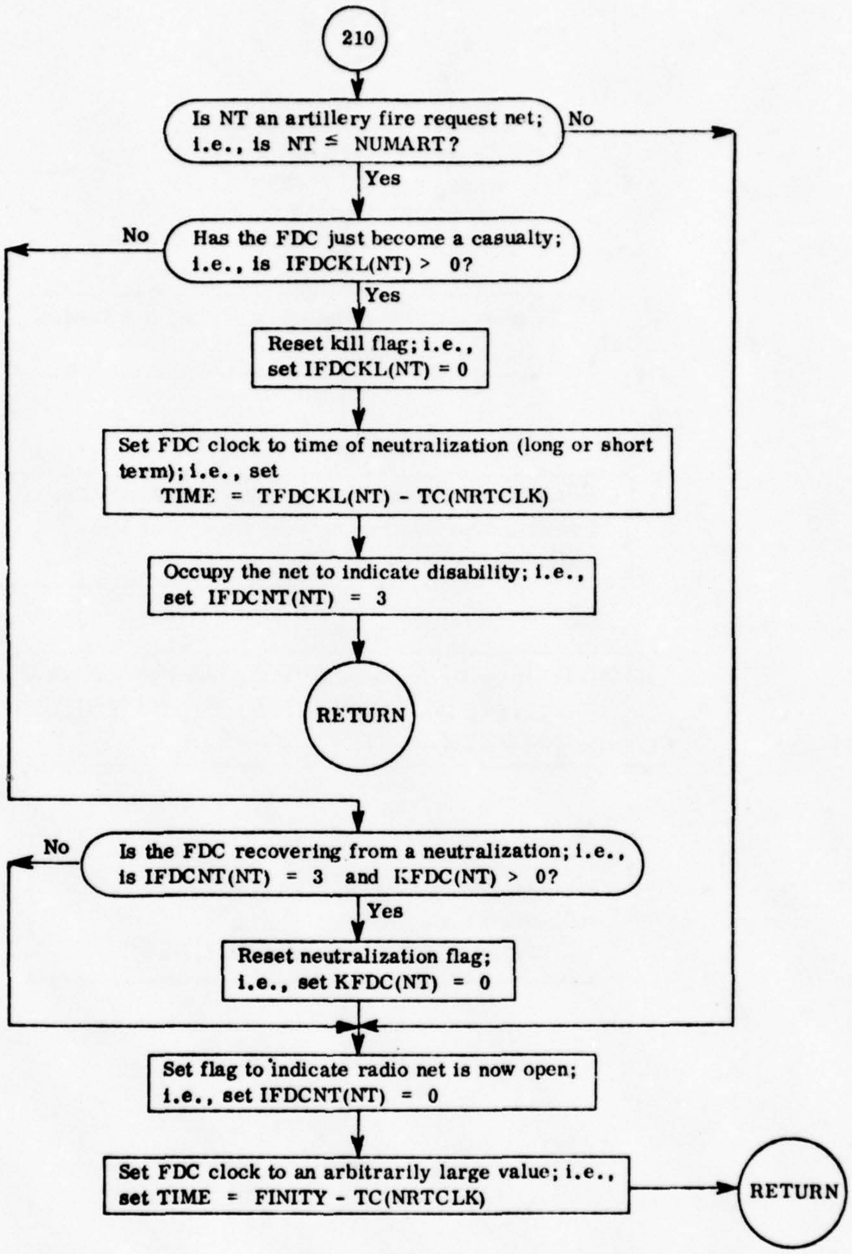
Subroutine AFDC: Continued



Subroutine AFDC: Continued



Subroutine AFDC: Continued



Subroutine AFDC: Continued

Subroutine AFO

PURPOSE: Subroutine AFO is designed to represent the activities of forward observers during target selection, fire request communication, and fire adjustment events.

CALLING SEQUENCE:

CALL AFO (TIME,NFO)

where

TIME = event time (output)
NFO = forward observer number (input).

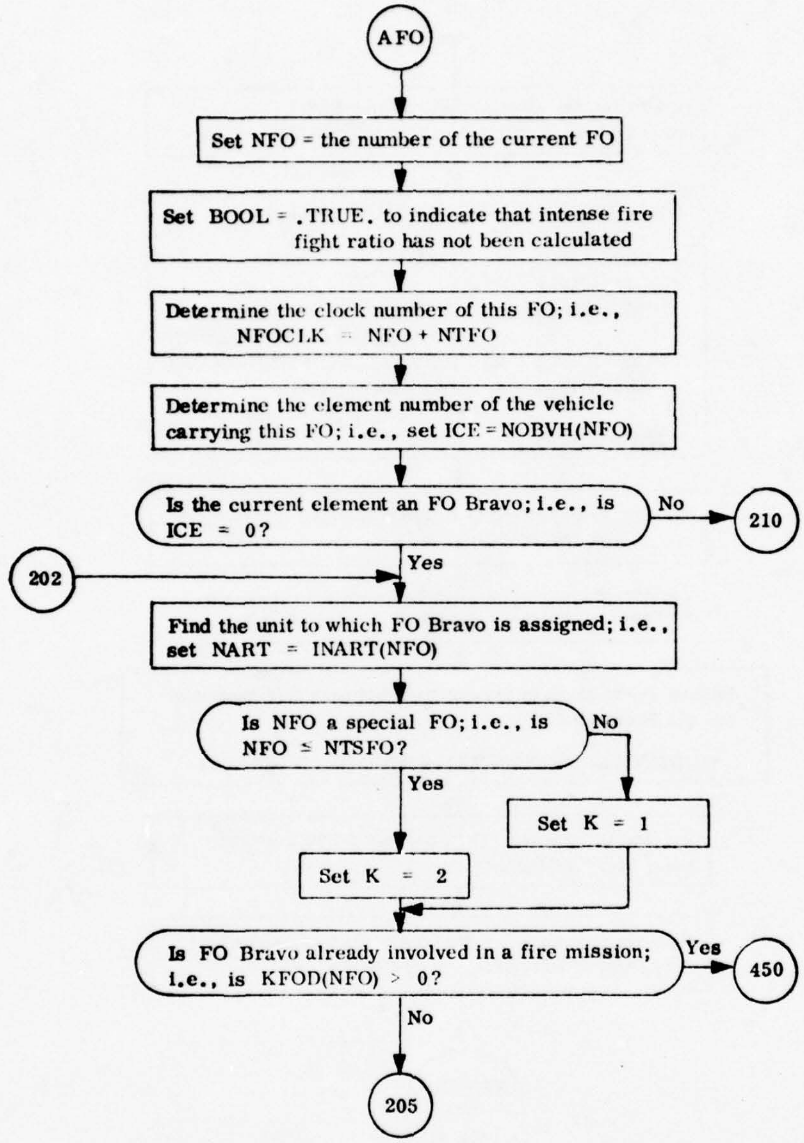
METHOD: See Chapter 2 of RF 2978 FR 71-3A (U).

COMMON AREAS REFERENCED:

AFODUZ	ITRIG	MNTRIG	SEQPAR
AMBDA	IWEAP	NCON	SIGCOM
CLOSE	JARMOR	NCONC	SIGCOR
DURC	KFDC	NFB	SIGPBR
DURFO	KFOD	NFOFR	SIGRBR
DURON	KILCRI	NFR	STRTIM
DURRL	KPAT	NFRMAX	TIMET1
DURSTL	KPATFO	NHPB	TIMET2
ECLOCK	KPATRN	NMISUN	TINIT
ENDMIS	KPRIOR	NOBVH	UCOM
EVBAR	KPTFRL	NRND	UCOREC
FRN1	KSEND	NRNDCN	UPRBRA
FRN2	KWAVAL	NRNDFR	URNBRA
ICECOM	LHICE	NSTHFF	WAITFO
IFBMIS	LKILL	NTELE	WAITMX
IFDCNT	LPREV	NTRANS	XCONC
IFRFL	LVENT	NUMBER	XD
IHAMO	LWCOD	NUMFIR	XDFO
IHDFMC	LWSYS	NUMTRG	XFRL
IHTARG	MAINPR	NVOLM	YCONC
IMIST	MDFAF	NXCONC	YD
INART	MESCON	NXFRL	YDFO
INTARG	MISFO	NXMFO	YFRL
INTRIG	MISFRL	ONDIST	
IPRIOR	MISTYP	ORESPN	
IPRIRR	MNTHRS	RCRIT	

SUBROUTINES REQUIRED:

APRIOR	RANND
CNFLCT	SELECA
ELOC	TRIG
FRANUD	WPRIOR
FSCMON	WRITFO
GETICE	WRTFRL



Subroutine AFO: Forward Observer Activities

205

Determine the weapon code of unit NART; i.e., set LWC = LWCOD(NART + NUMELE)

Obtain fire-request data for FO Bravo; i.e.,

XDFO(NFO) = -8000
YDFO(NFO) = -8000
KPATFO(NFO) = 1
IPRIOR(NFO) = UPRBRA(K, LWC) +N(0, 1)*SIGPBR(K, LWC)
DURFO(NFO) = DURC(K, LWC)+TC(NFOCLK)
NRND(NFO) = URNBRA(K, LWC) +N(0, 1)*SIGRBR(K, LWC)
KSEND(NFO) = LWC

Set flag to indicate FO Bravo has a fire mission; i.e., set MISFO(NFO) = 1

Obtain U from a uniform (0, 1) distribution

Update FO's clock to reflect time between fire requests for FO Bravo; i.e.,

TC(NFOCLK)=TC(NFOCLK)-AMBDA(K, LWC)* ln(U)

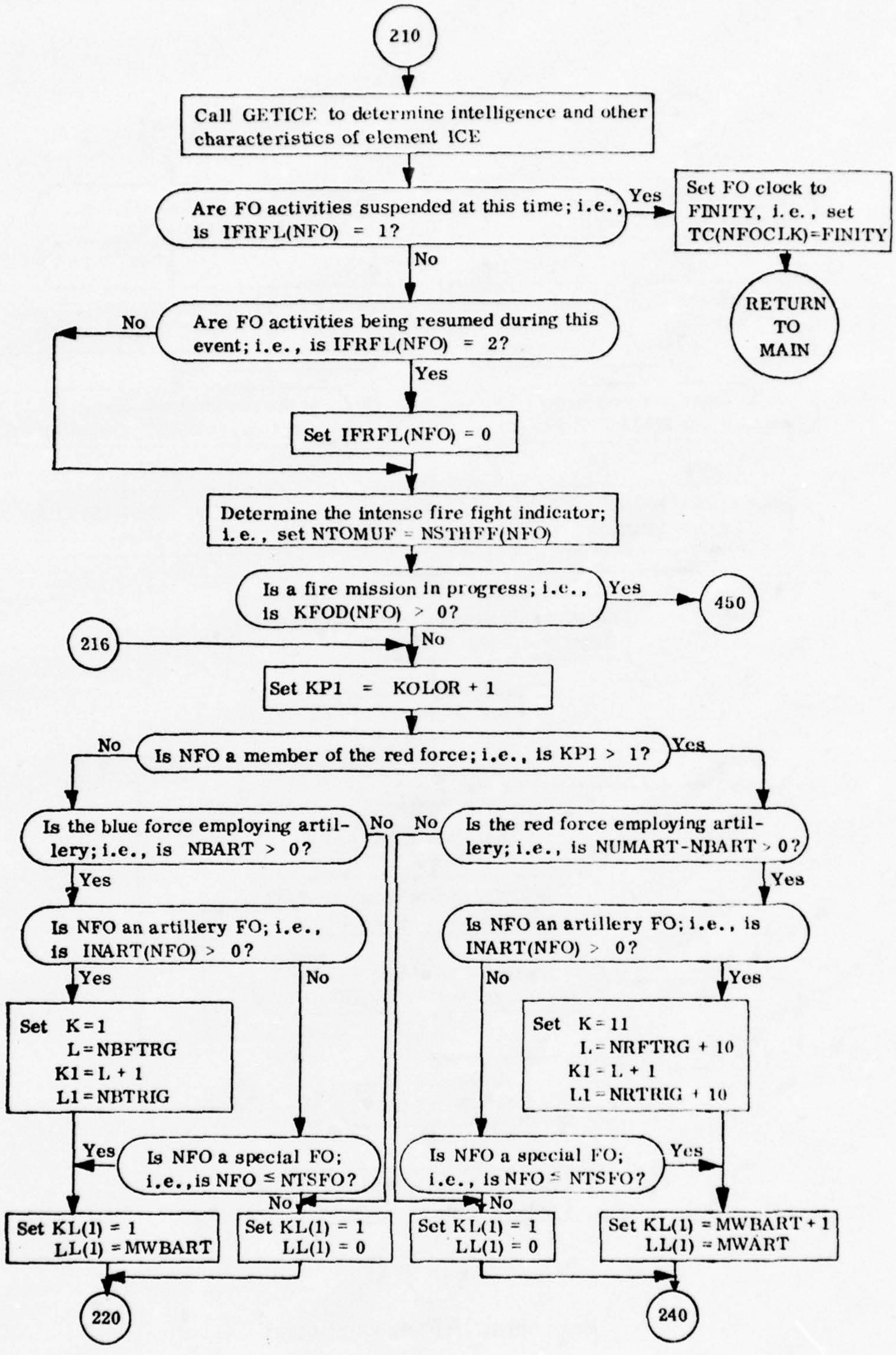
Set flag to indicate FO Bravo has a fire request; i.e., set KFOD(NFO) = 1

Record time at which current target was selected; i.e., set TINIT(NFO) = TC(NFOCLK)

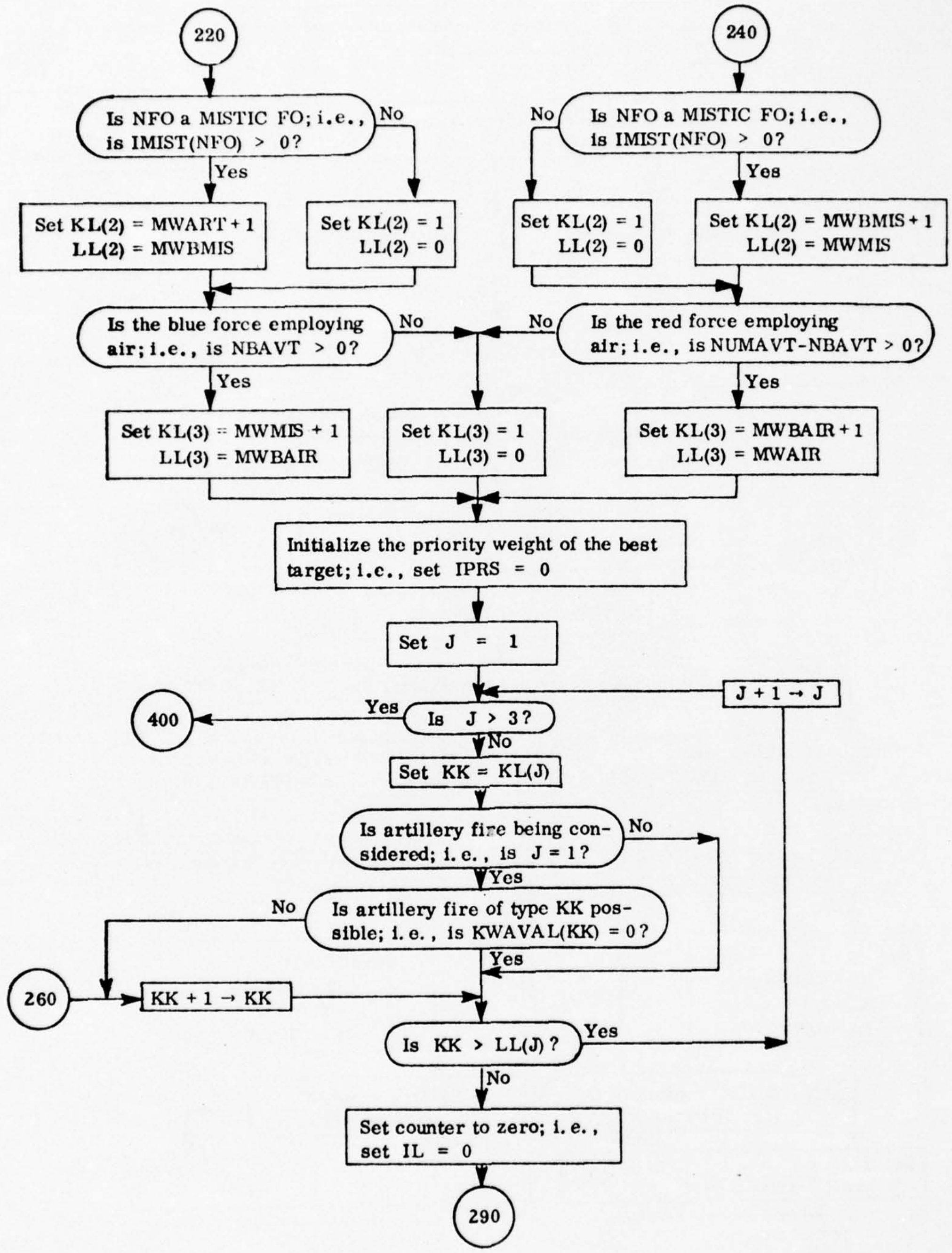
RETURN TO MAIN

B-24

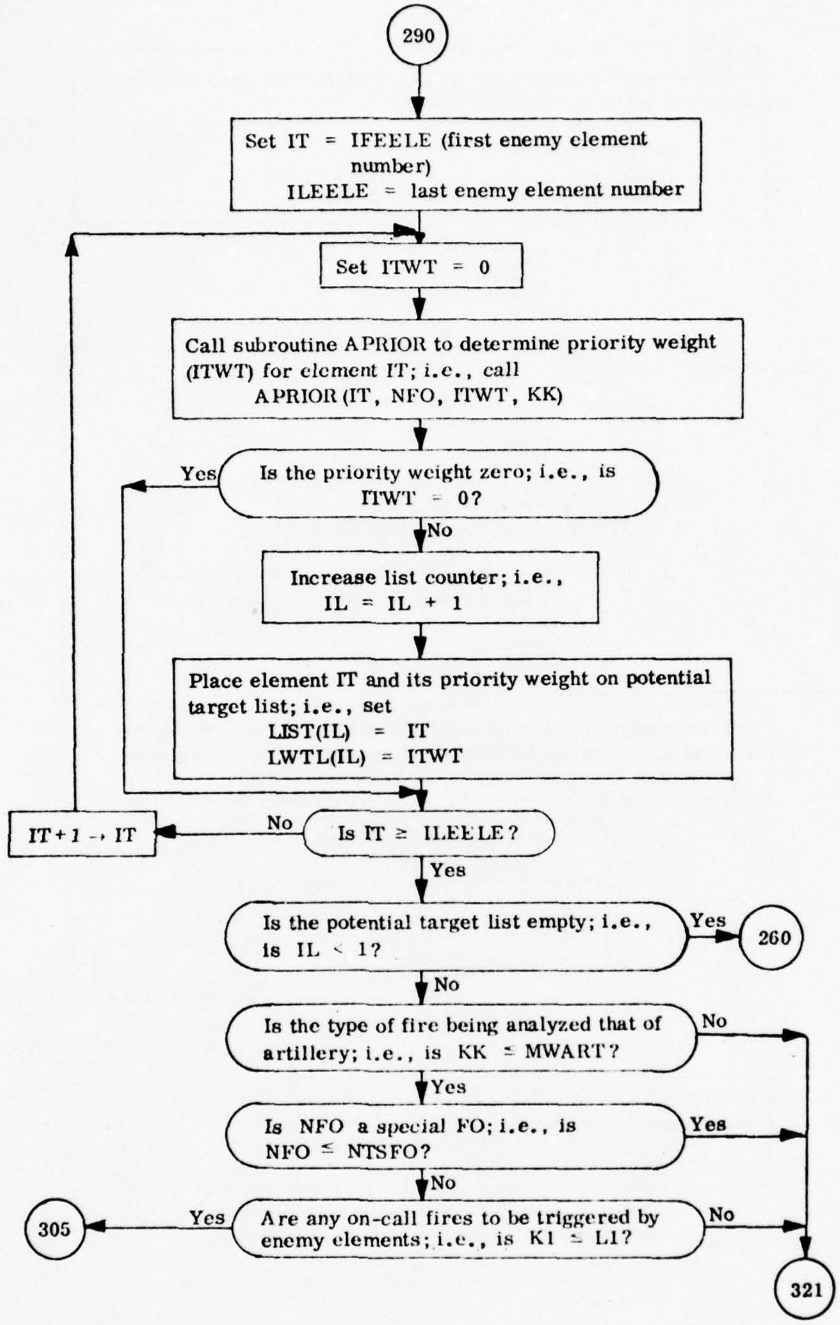
Subroutine AFO: Continued



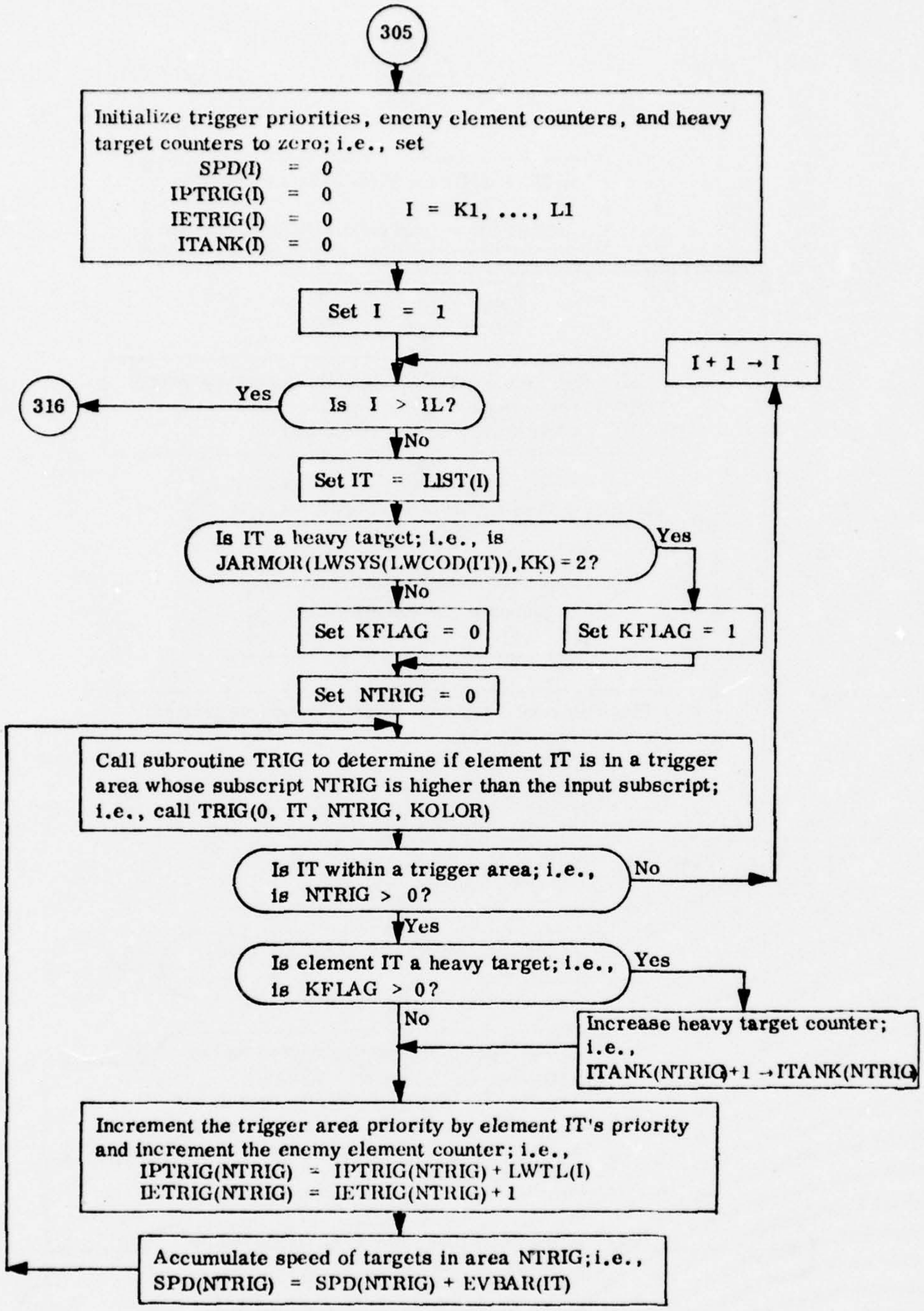
Subroutine AFO: Continued



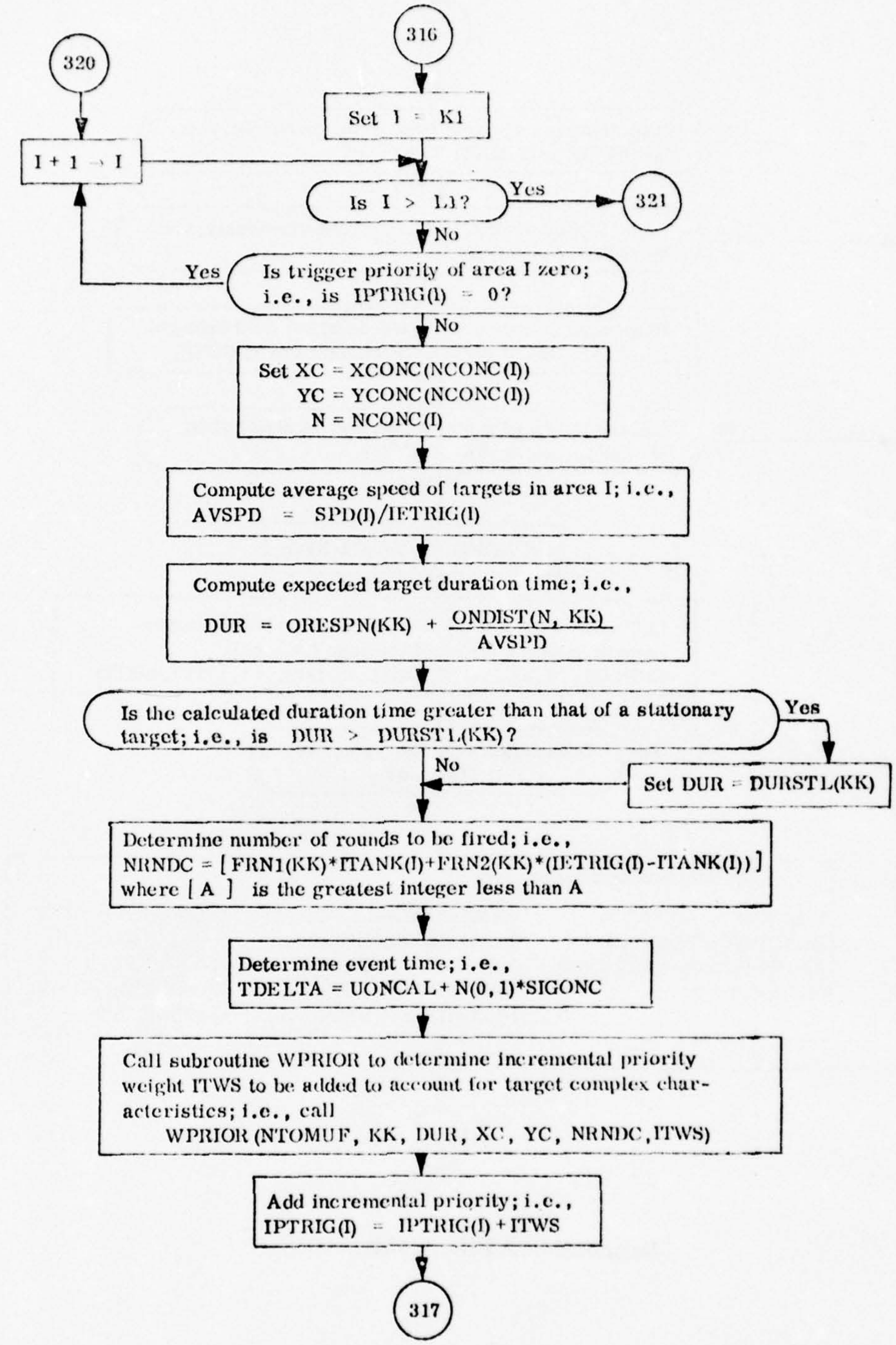
Subroutine AFO: Continued



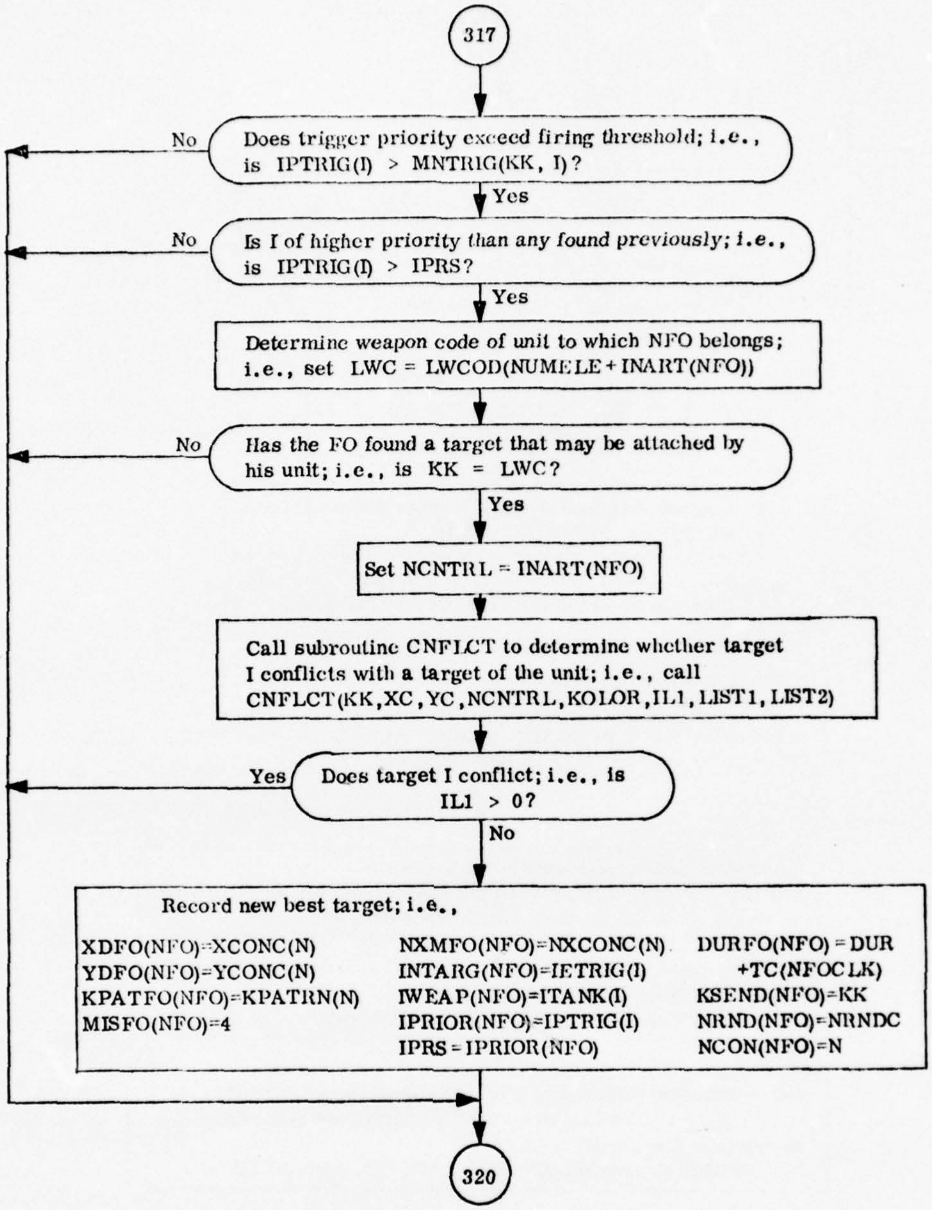
Subroutine AFO: Continued



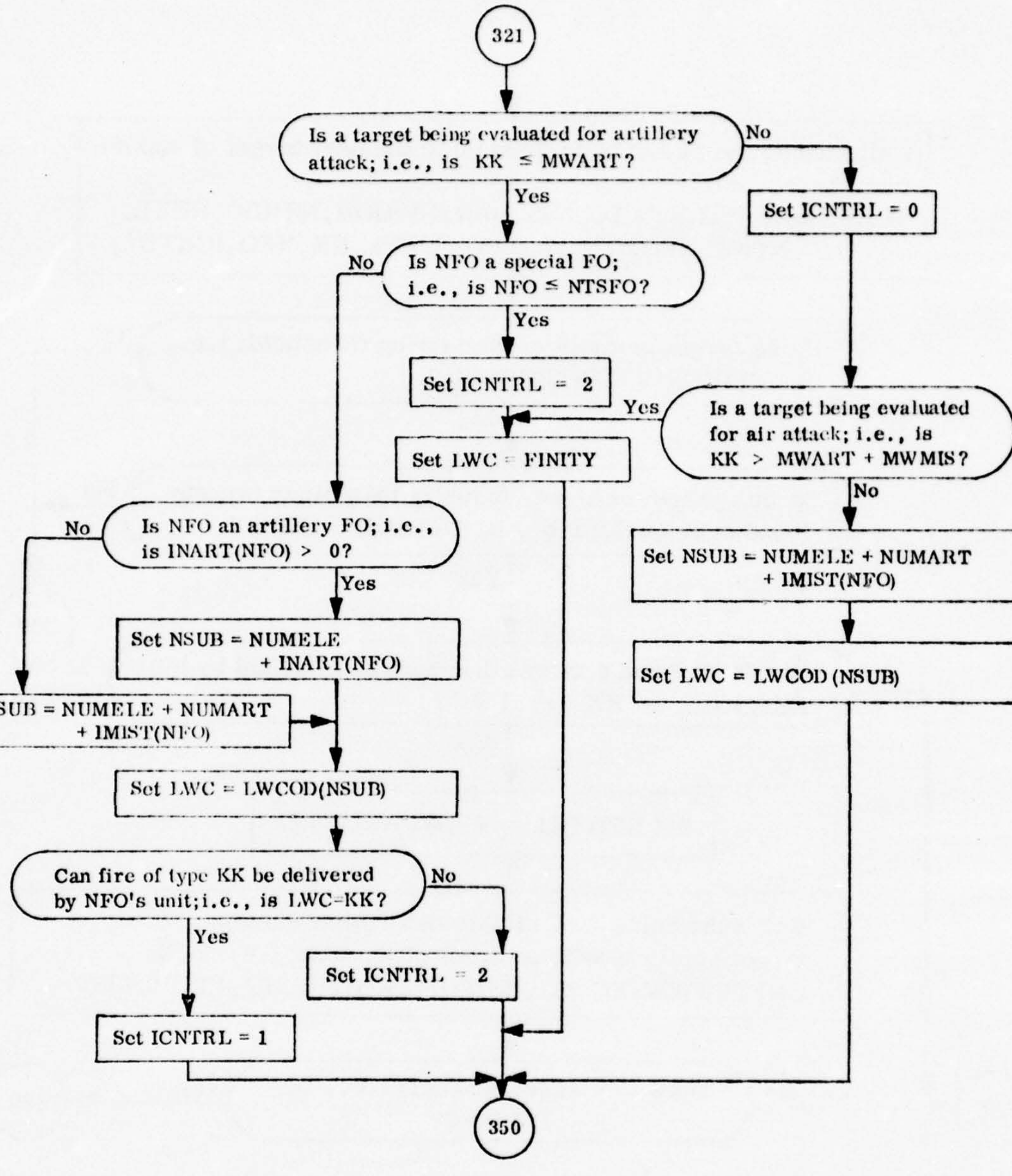
Subroutine AFO: Continued



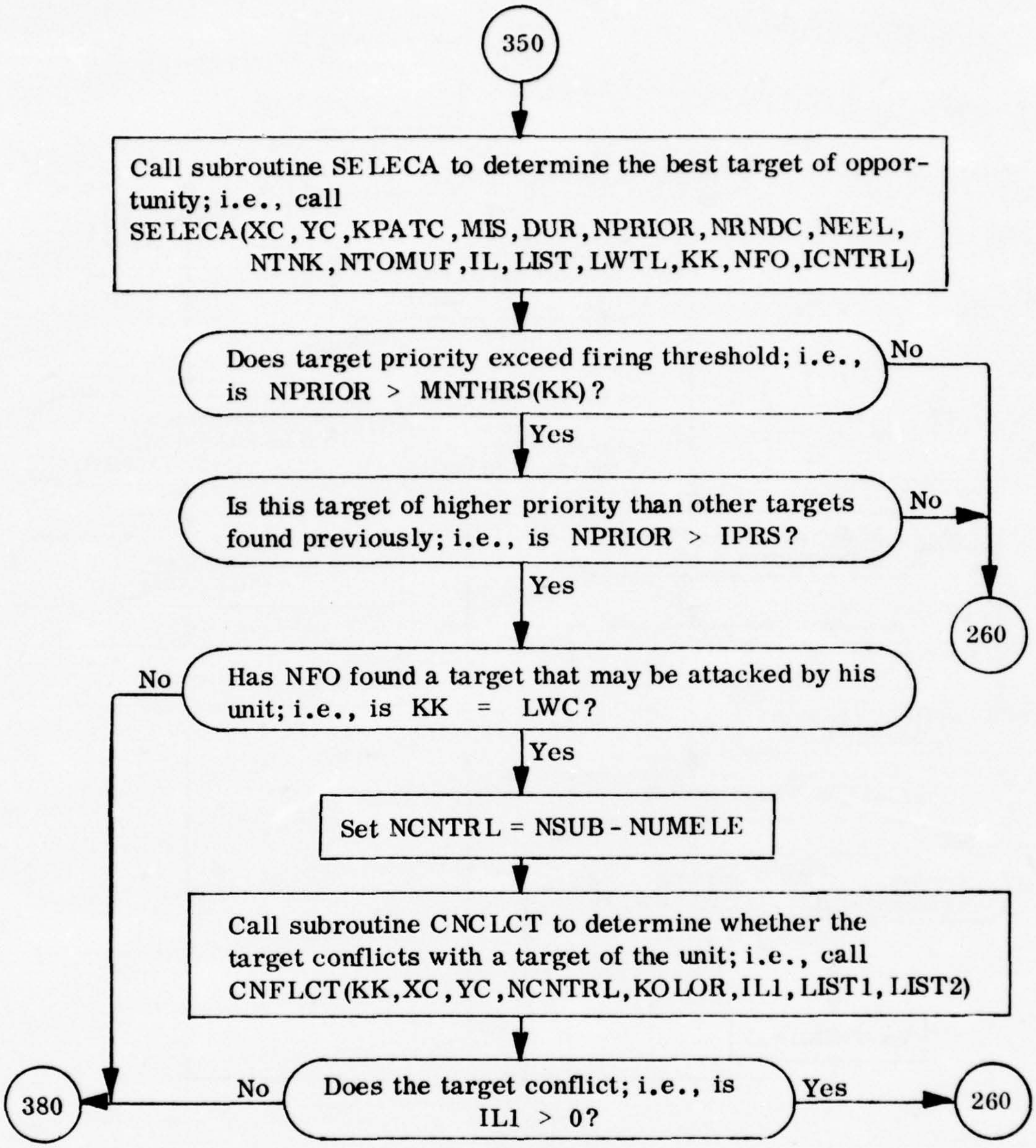
Subroutine AFO: Continued



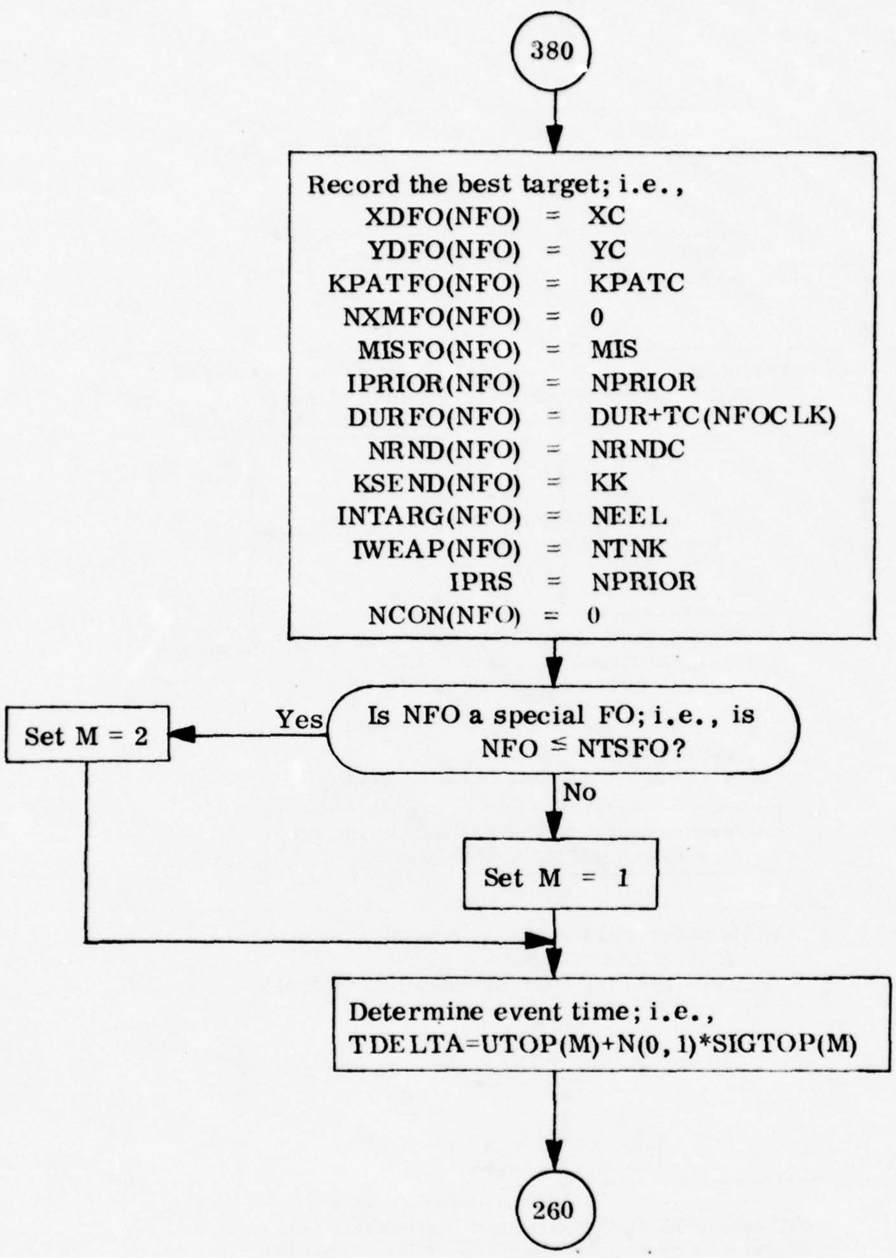
Subroutine AFO: Continued



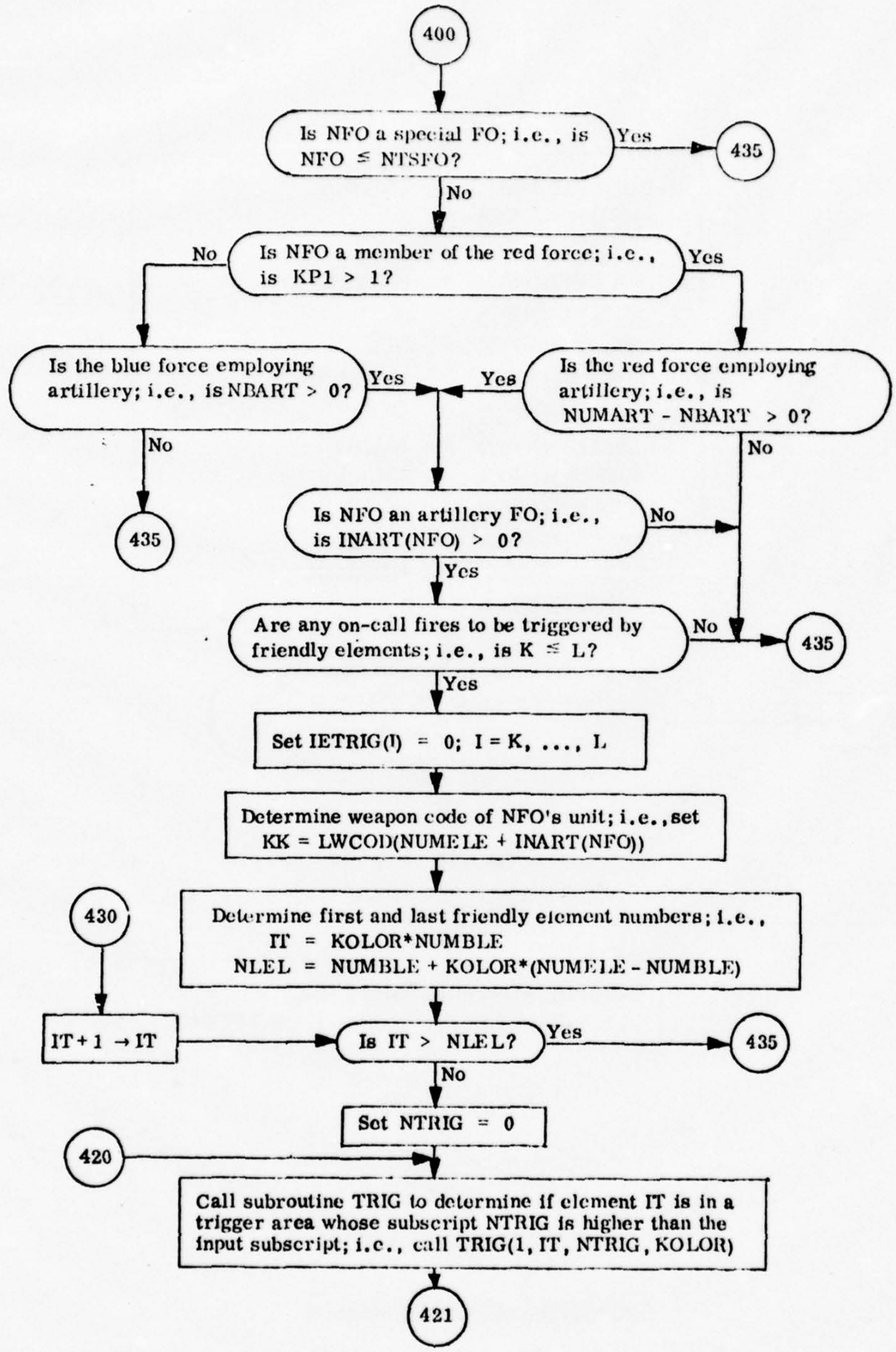
Subroutine AFO: Continued



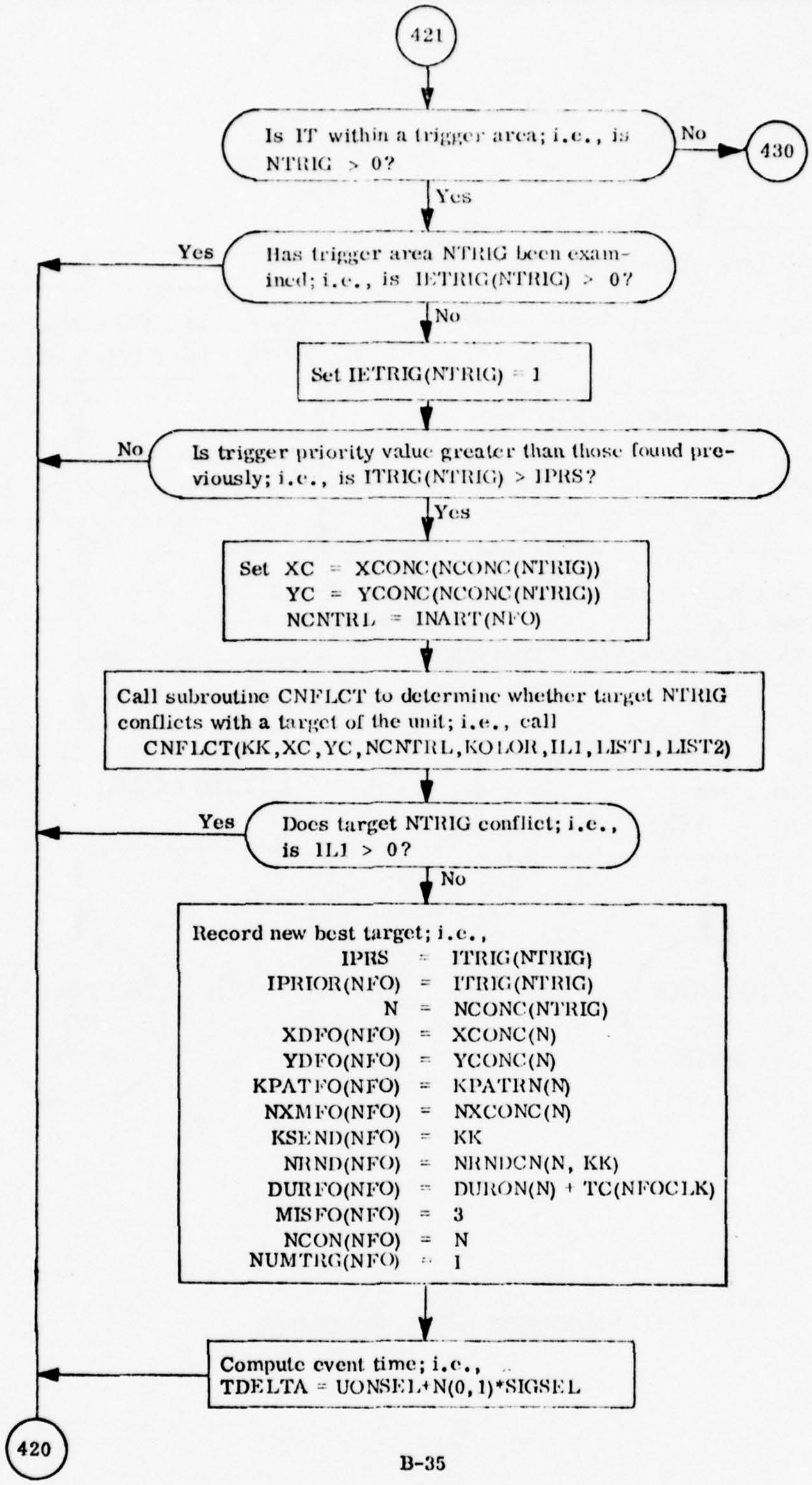
Subroutine AFO: Continued



Subroutine AFO: Continued

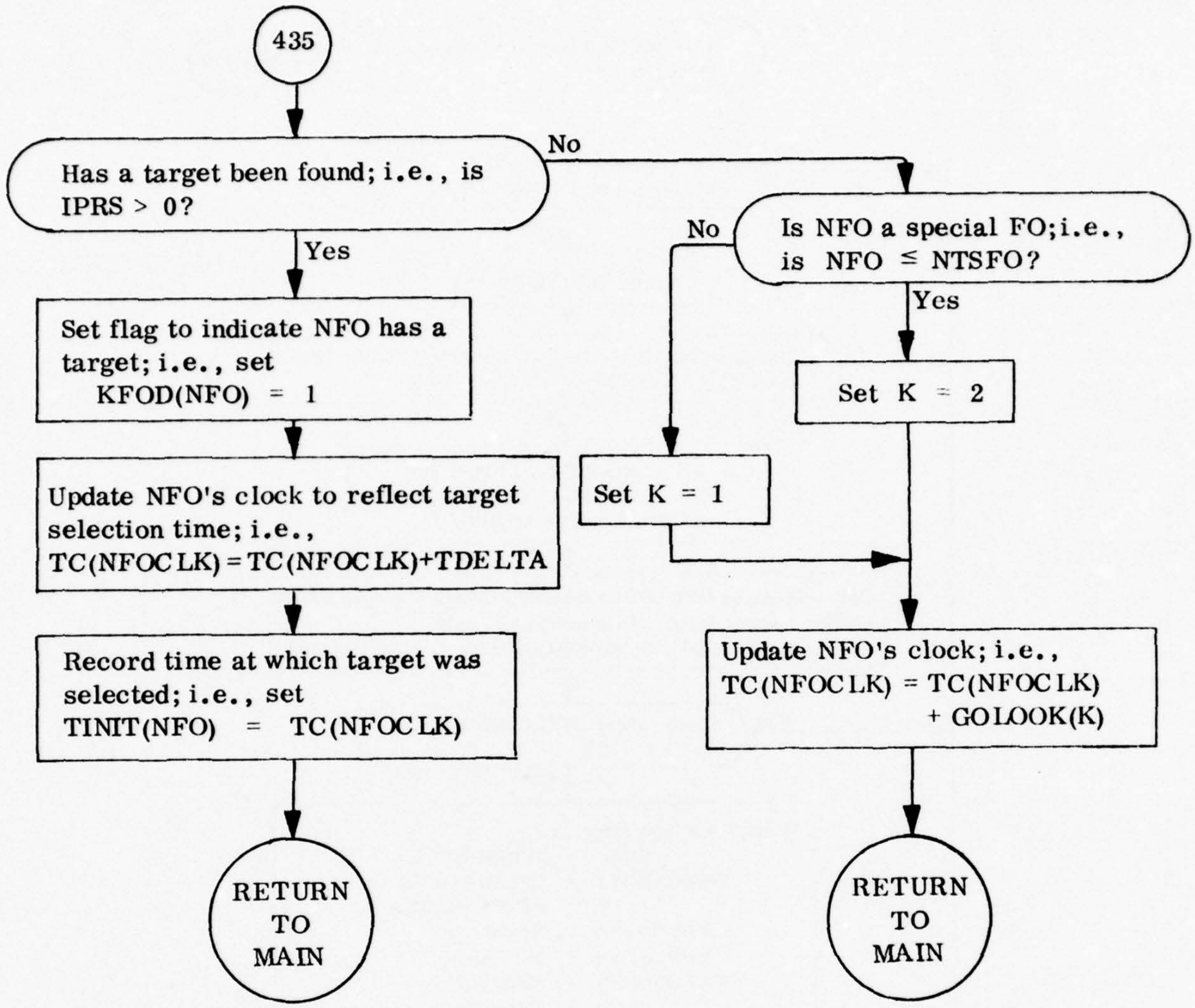


Subroutine AFO: Continued

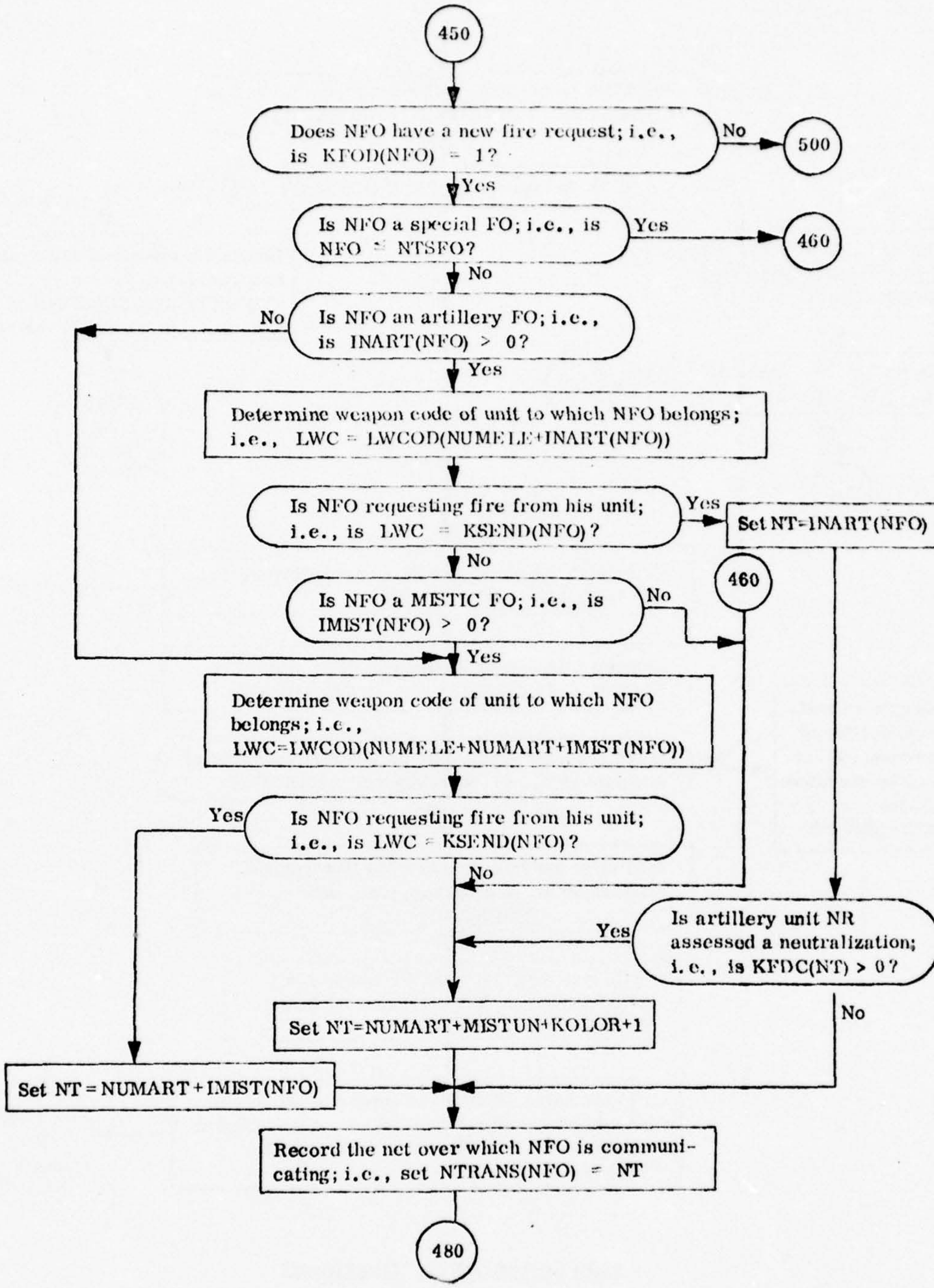


B-35

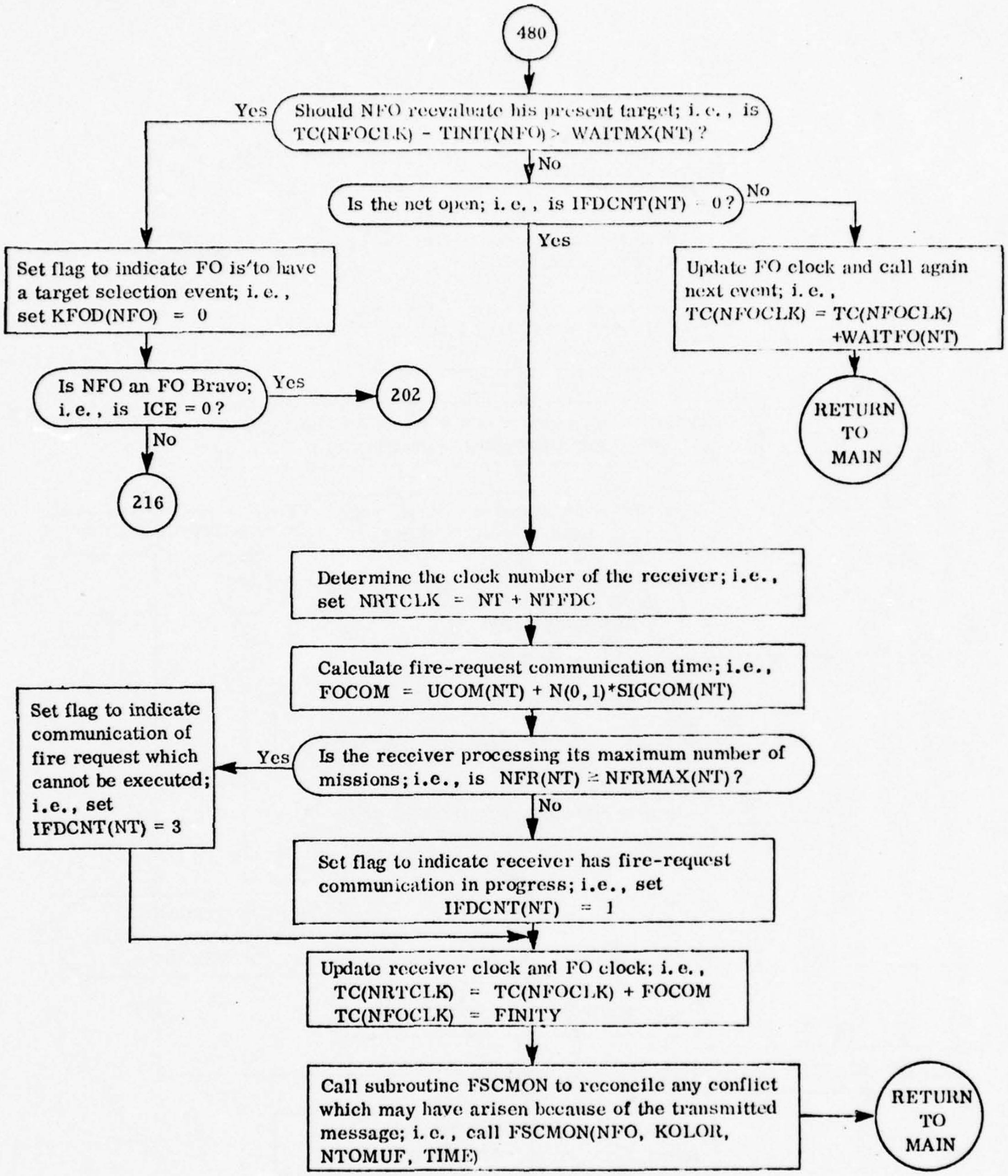
Subroutine AFO: Continued



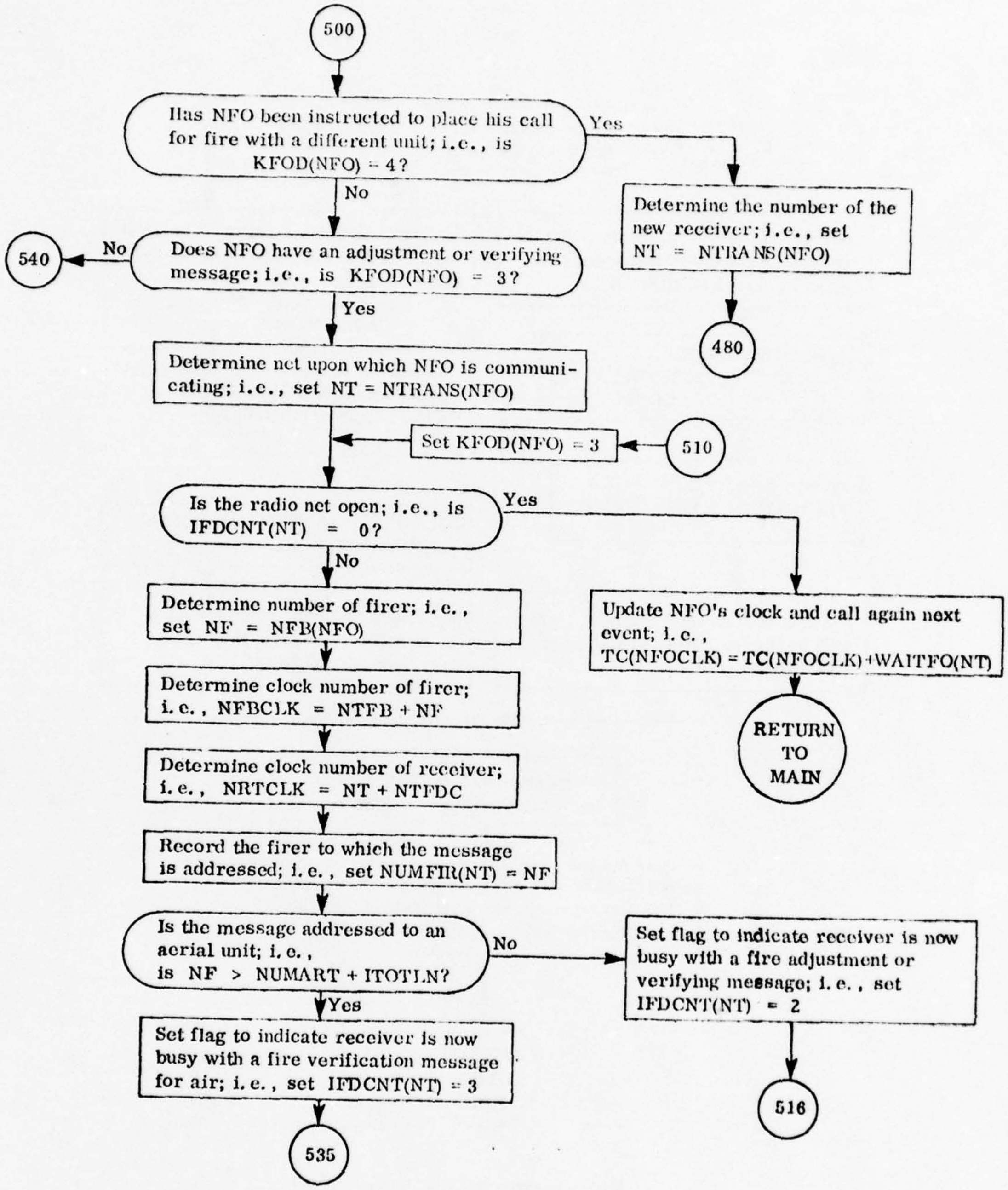
Subroutine AFO: Continued



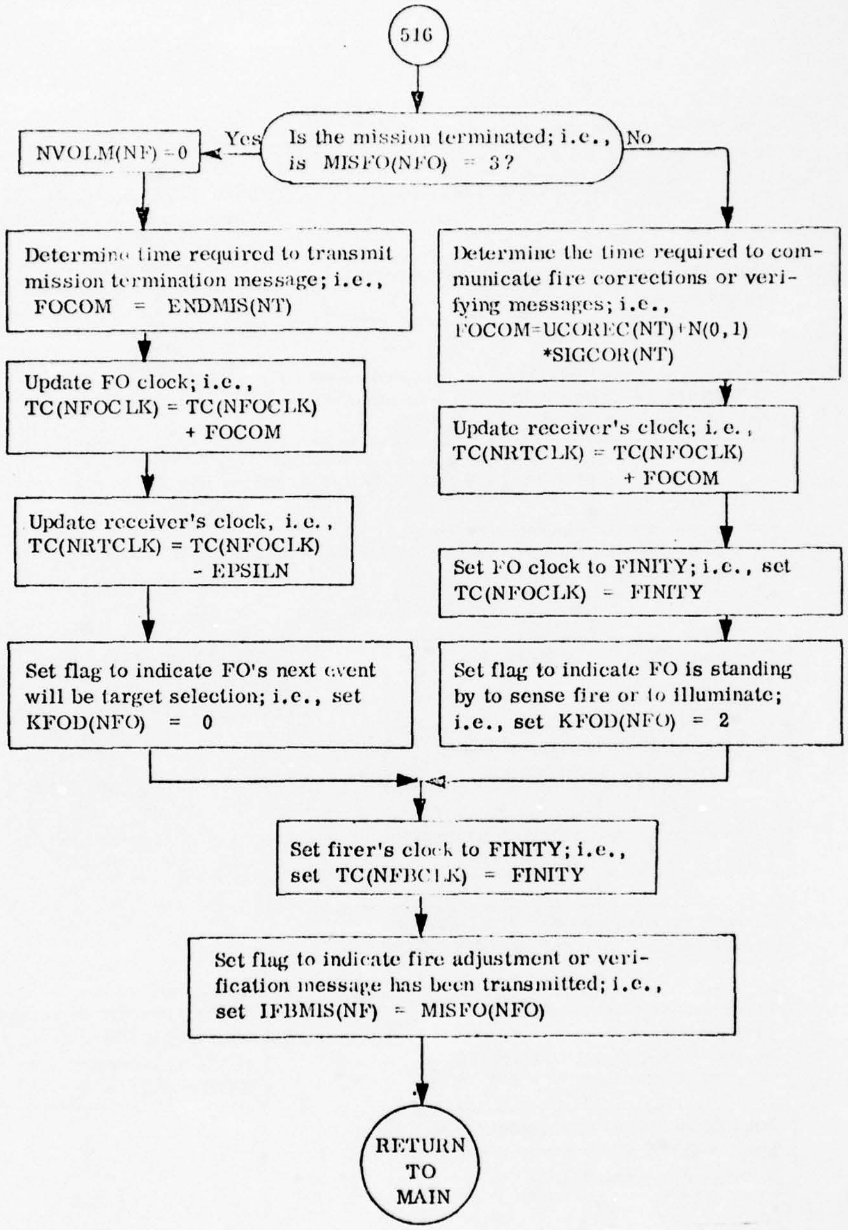
Subroutine AFO: Continued



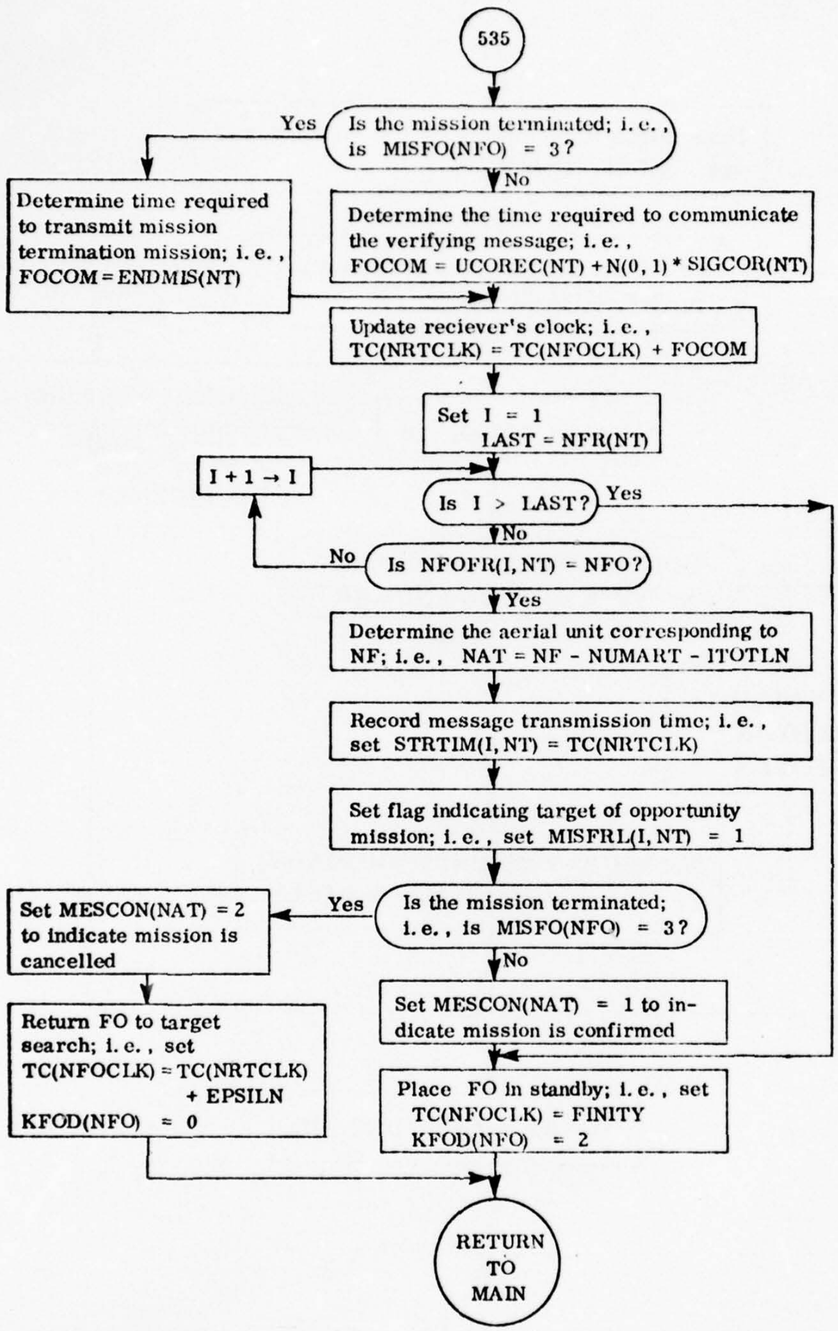
Subroutine AFO: Continued



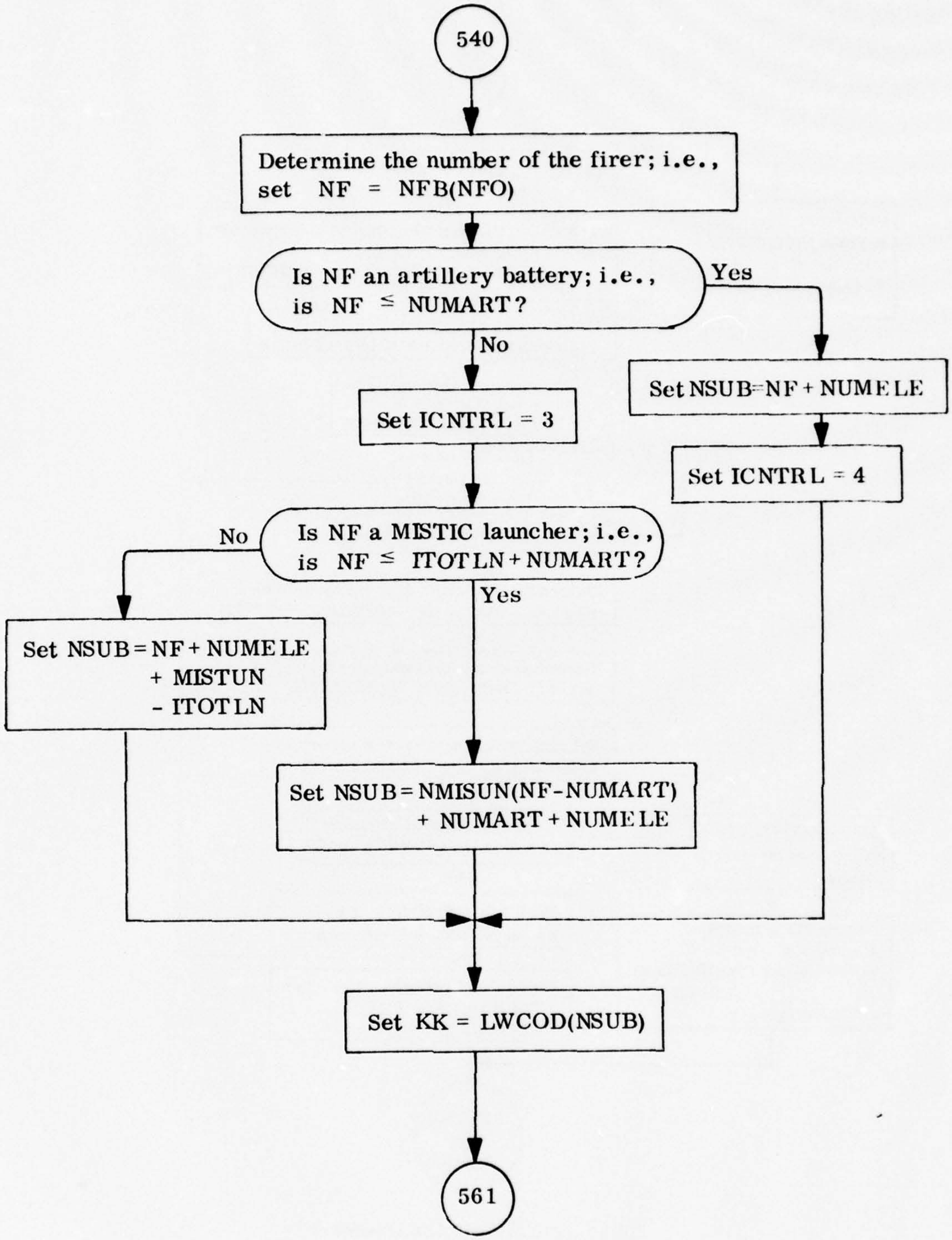
Subroutine AFO: Continued



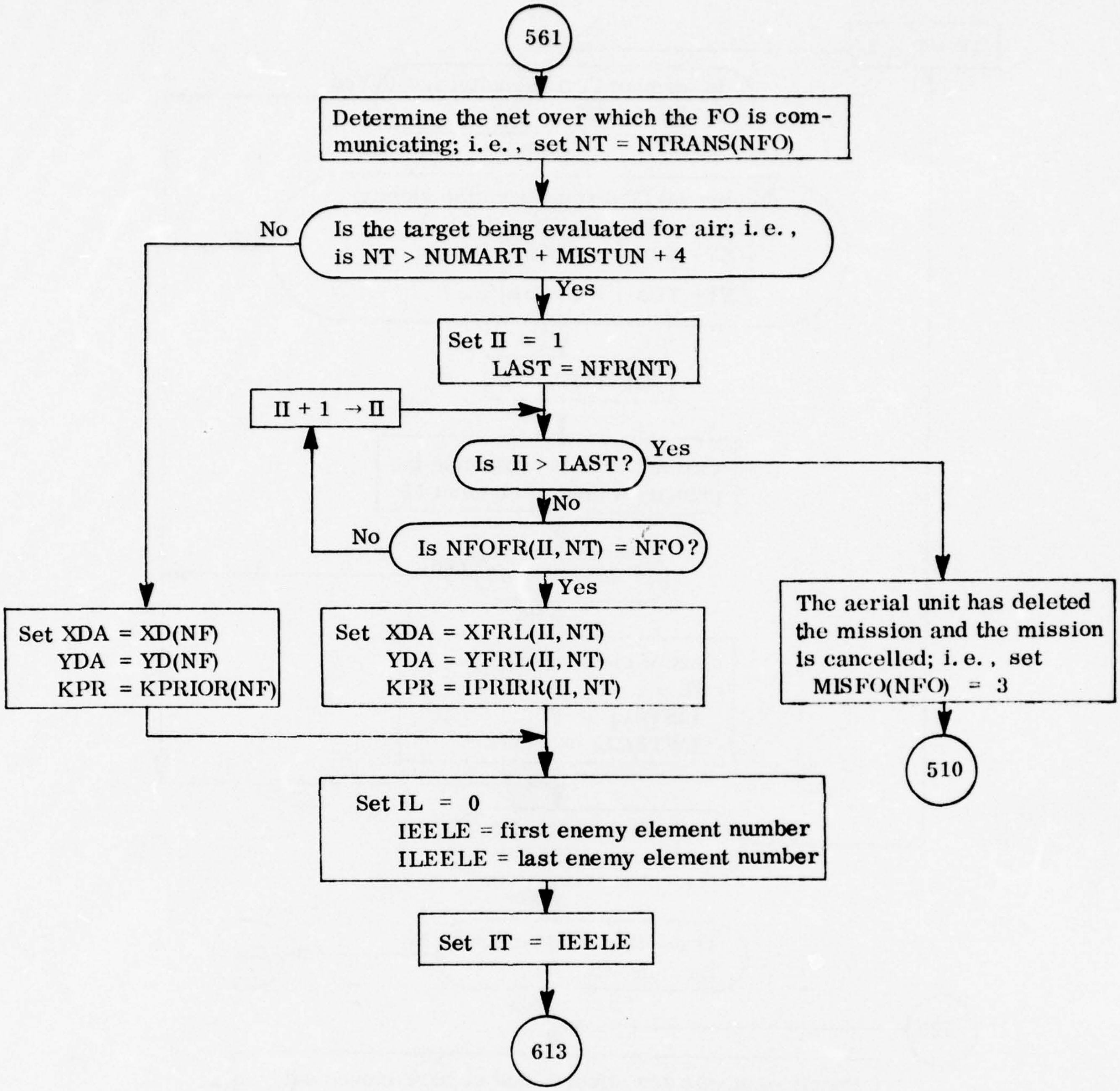
Subroutine AFO: Continued



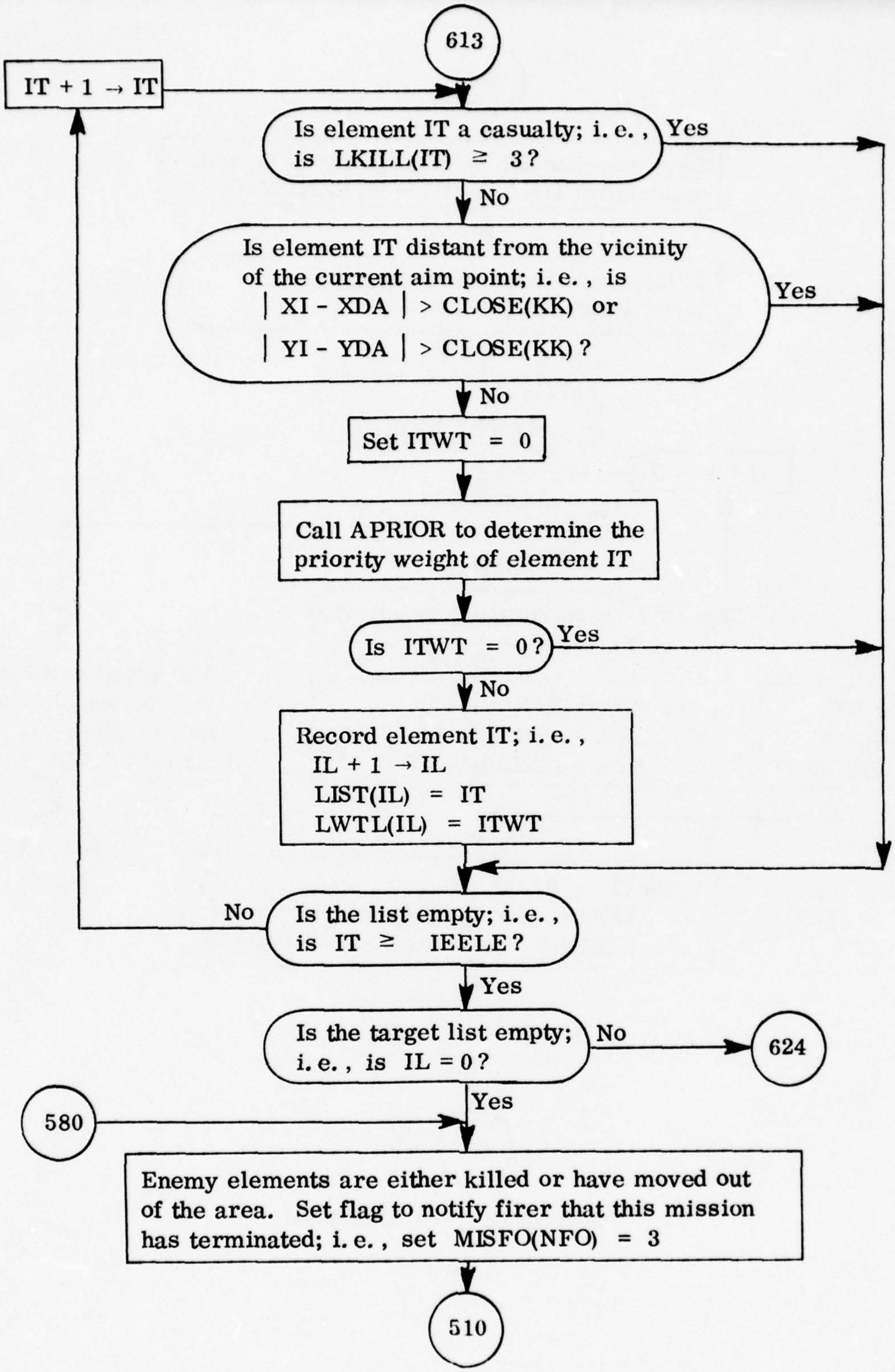
Subroutine AFO: Continued



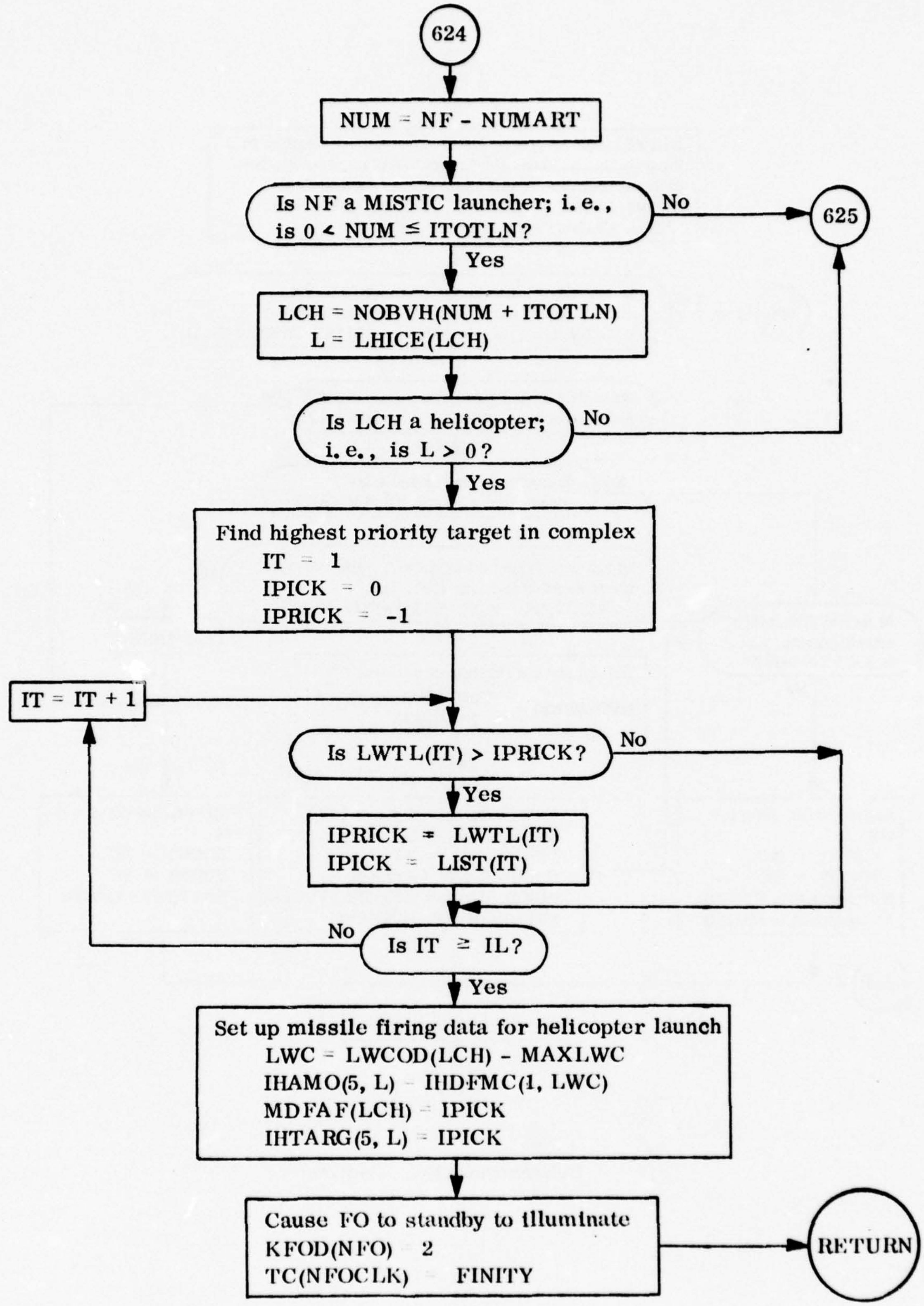
Subroutine AFO: Continued



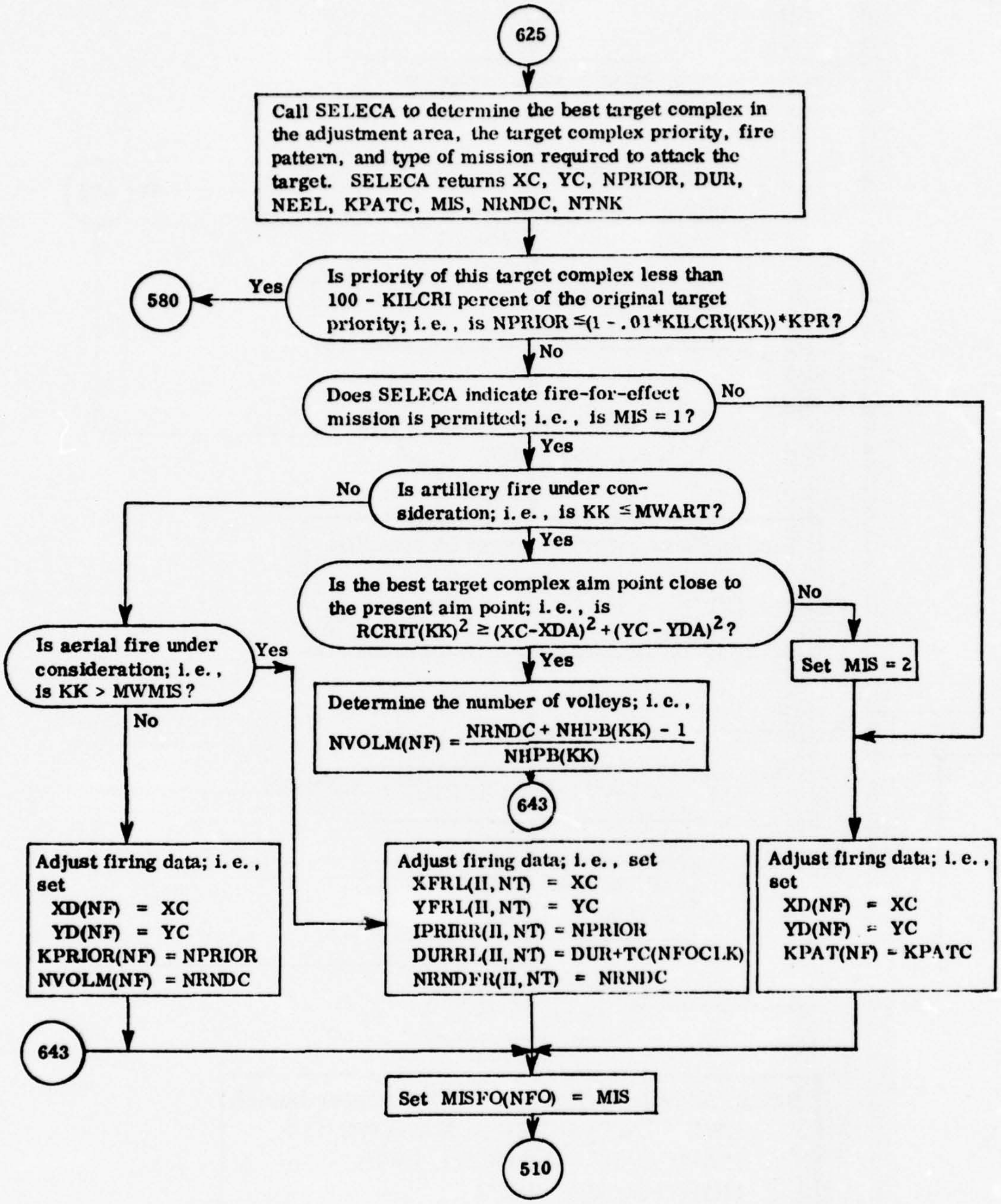
Subroutine AFO: Continued



Subroutine AFO: Continued



Subroutine AFO: Continued



Subroutine AFO: Continued

Subroutine AFSC

PURPOSE: Subroutine AFSC is designed to represent the active operations of the fire support coordinator element associated with fire support mission assignment.

CALLING SEQUENCE:

CALL AFSC (NT, TIME)

where

NT = radio net number
TIME = event time increment

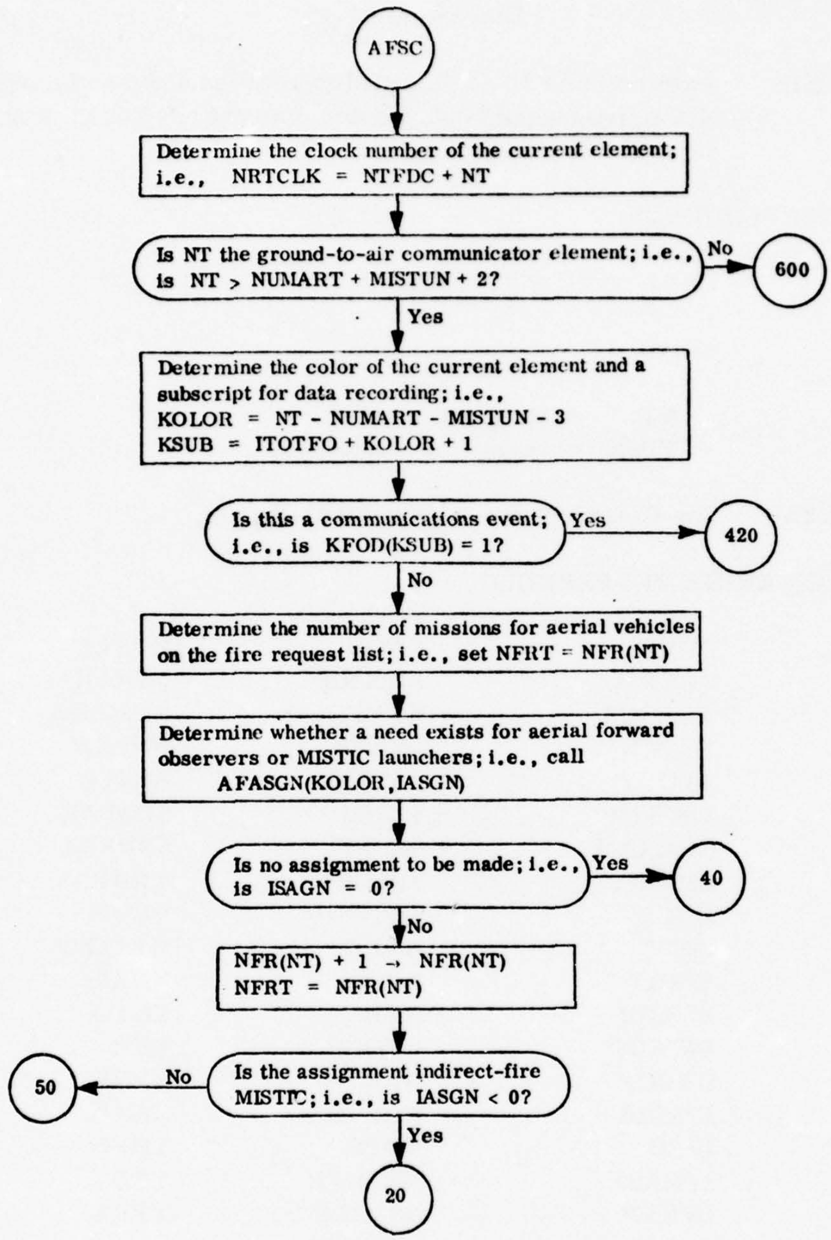
METHOD: See Chapter 2 of RF 2978 FR 71-3A (U).

COMMON AREAS REFERENCED:

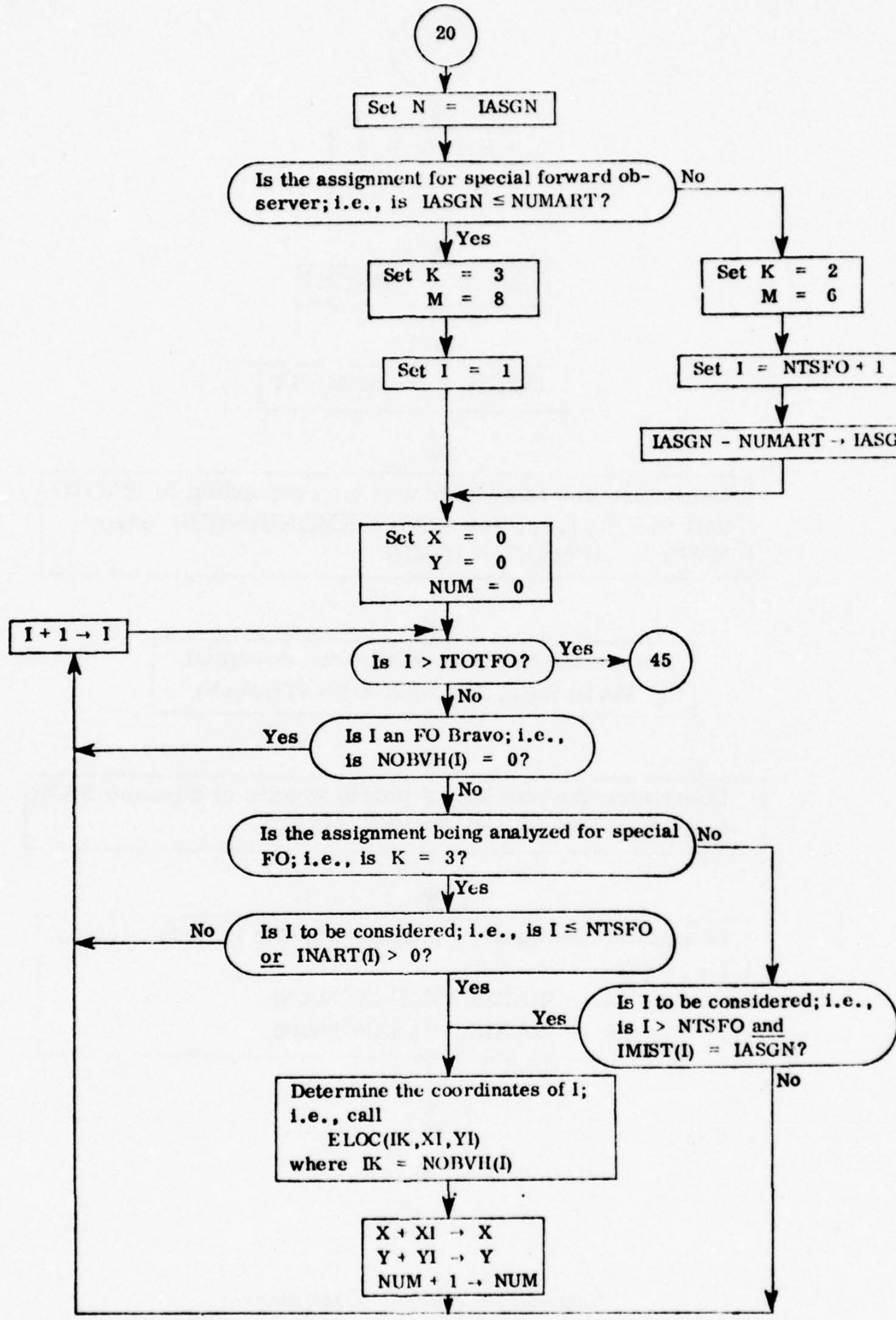
DURFO	KFOD	NTELE
DURRL	KMANN	NTRANS
ECLOCK	KPTFRL	NUMBER
ELOCX	KSEND	NXFRL
ELOCY	LFUNC	REJTIM
FSCTIM	LNUM	SEQPAR
GOEXAM	MISFO	SIGCOM
IFDCNT	MISFRL	TINIT
IFRFL	MISTYP	UCOM
IMIST	NAXIS	WAITFO
INART	NCON	XAXIS
INPRIR	NFB	XDFO
INTARG	NFOFR	XFB
IPRIOR	NFR	XFRL
IPRIRR	NOBVH	YAXIS
IREQ	NPTS	YDFO
ITMASS	NRNDFR	YFB
IWEAP	NSTHFF	YFRL
KFDC	MESARR	

SUBROUTINES REQUIRED:

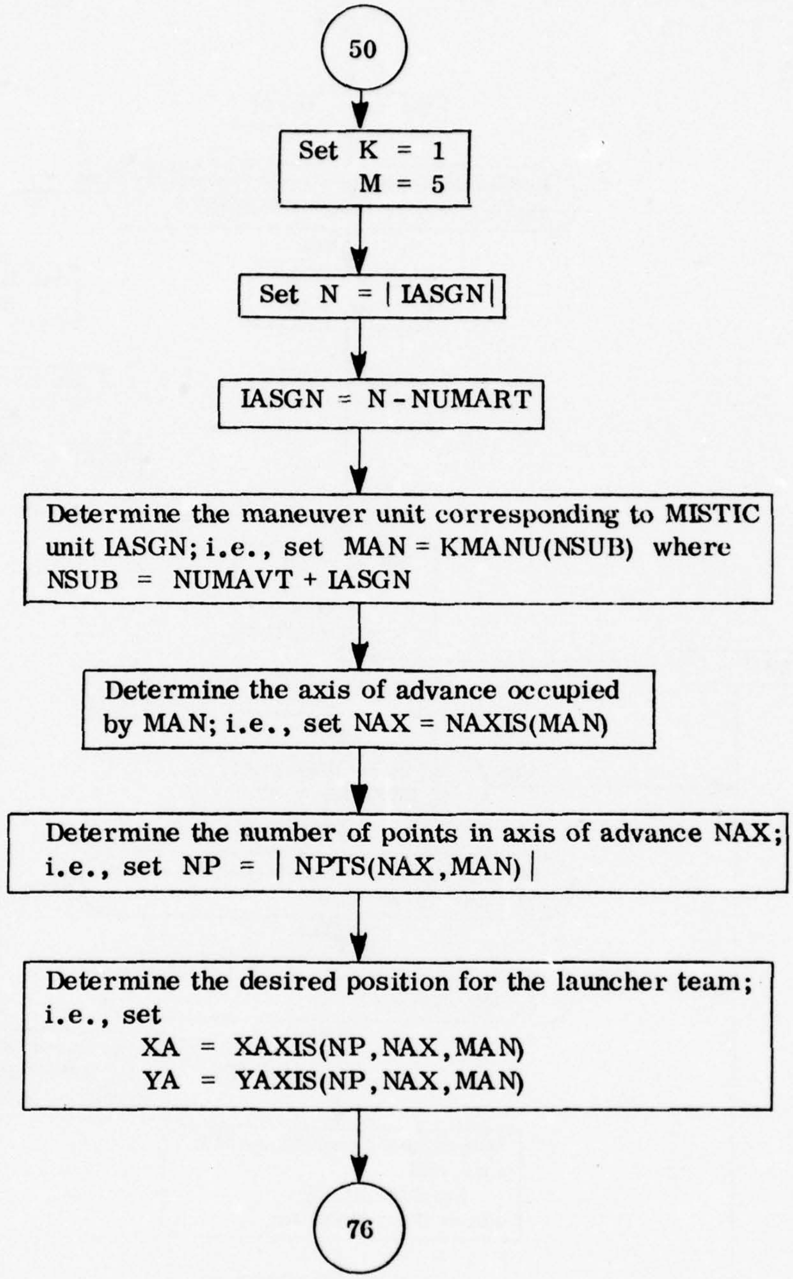
AFASGN	ELOC	REASFO
AIRSEL	ERROR	REVAL
AVAIL	FSCVAL	
CNFLCT	RANND	



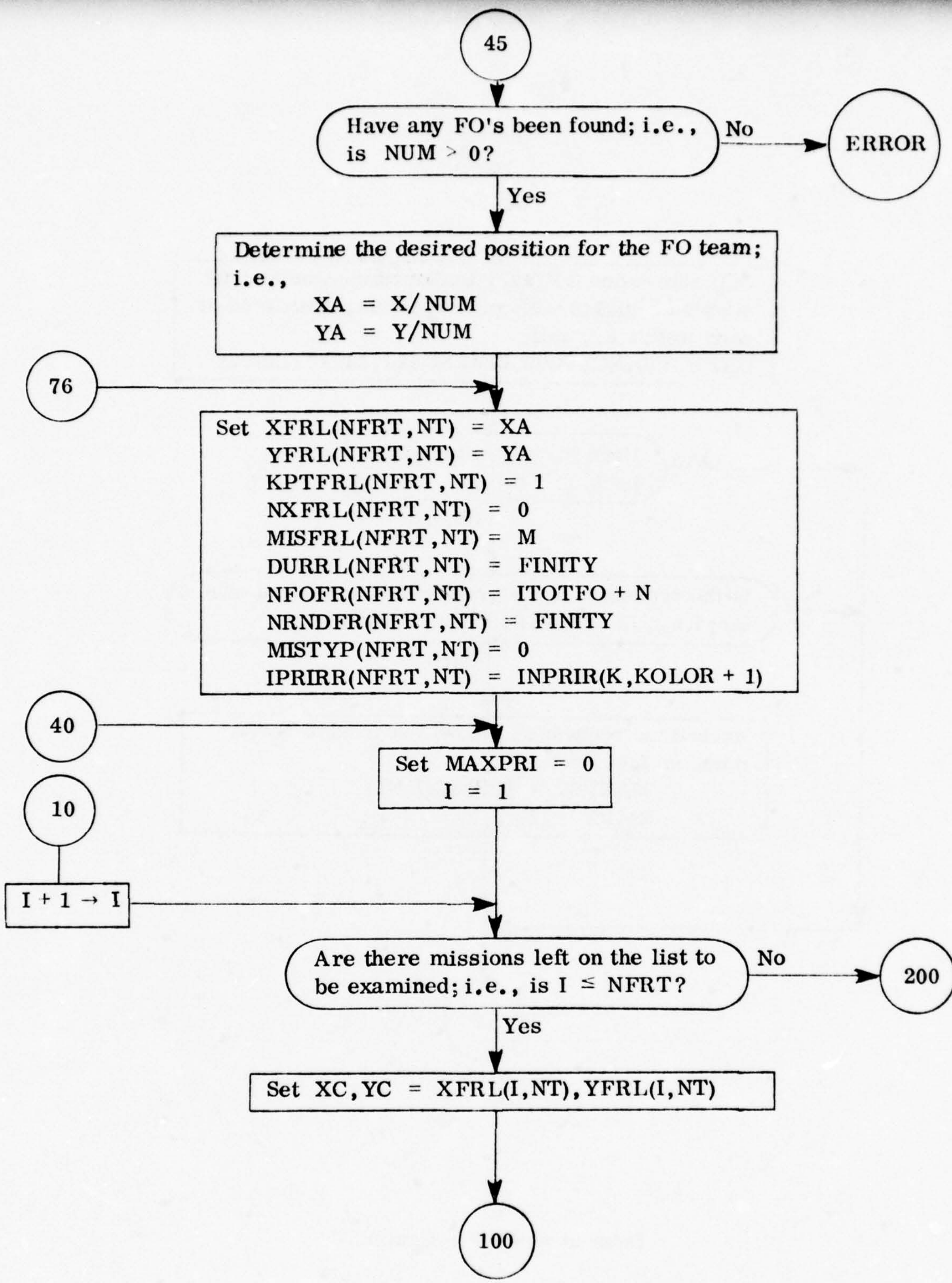
Subroutine AFSC: Fire Support Coordinator Activities



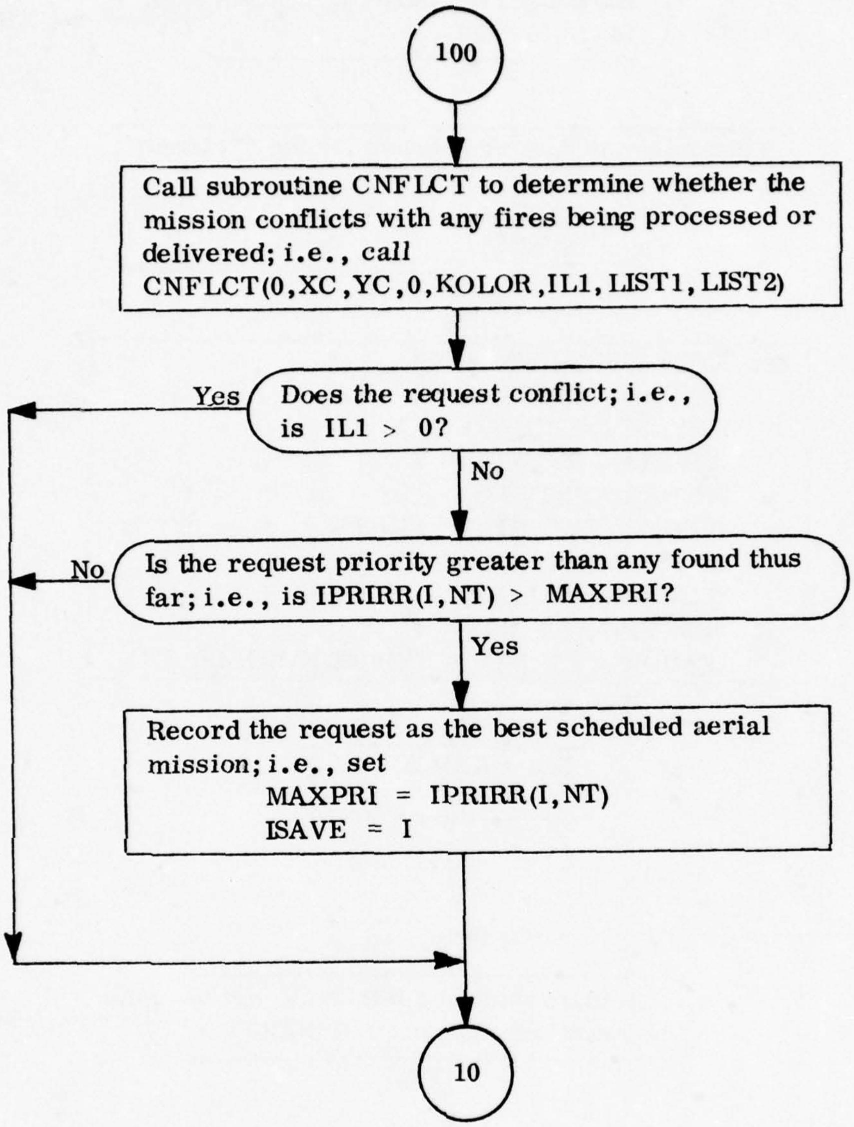
Subroutine AFSC: Continued



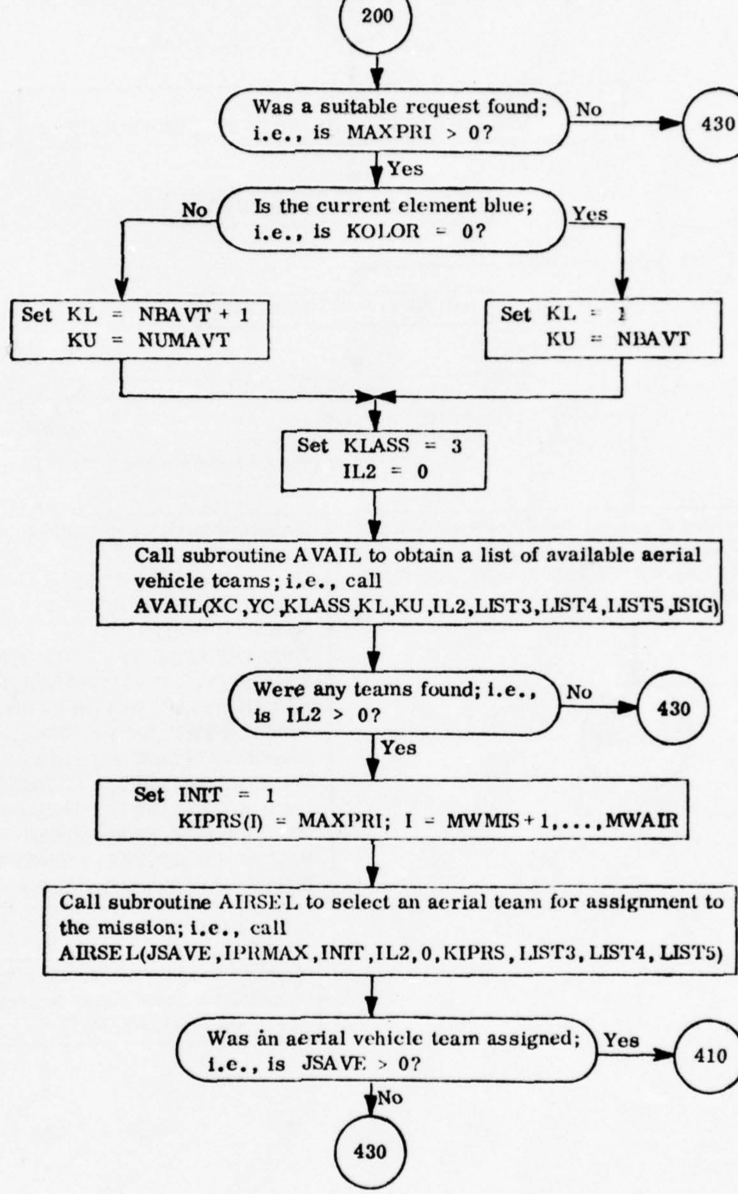
Subroutine AFSC: Continued



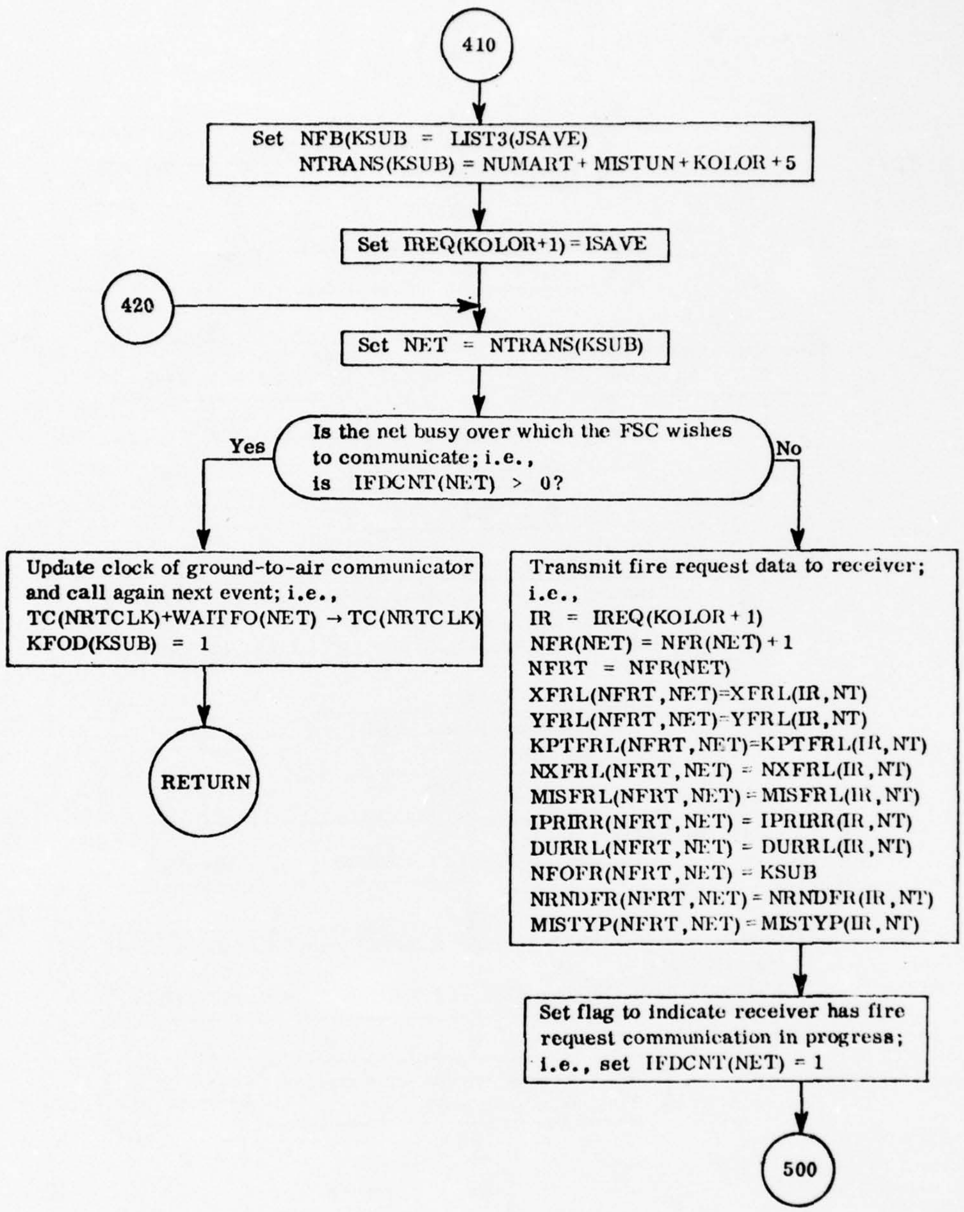
Subroutine AFSC: Continued



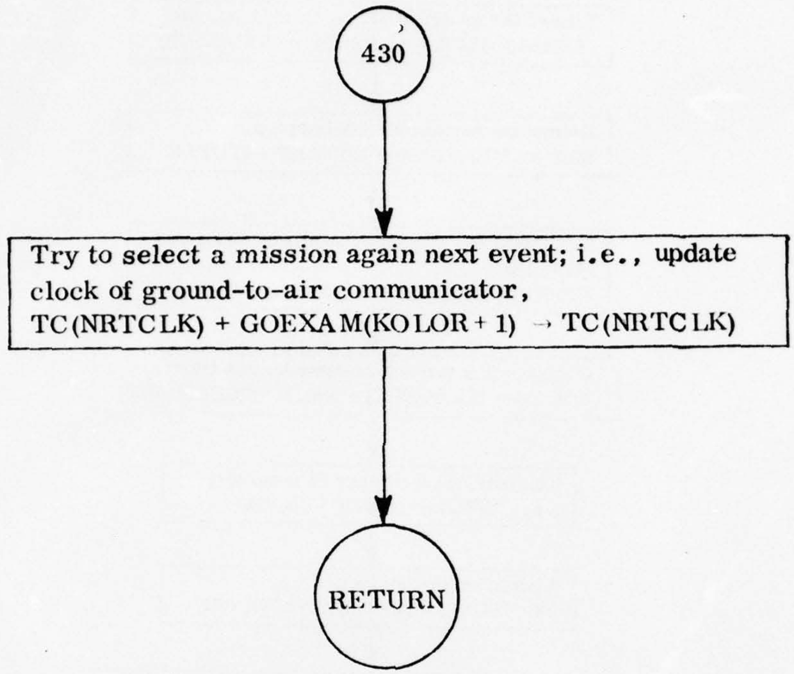
Subroutine AFSC: Continued



Subroutine AFSC: Continued



Subroutine AFSC: Continued



Subroutine AFSC: Continued

500

Record the assigned aerial team; i.e., set ITMASS(KOLOR + 1, NFRT) = NFB(KSUB)

↓

Determine aerial unit number; i.e., NAT = NFB(KSUB) - NUMART - ITOTLN

↓

Increment the messages awaiting aerial unit NAT; i.e., MESARR(NAT) = 1 + MESARR(NAT)

↓

Compute fire request communication time; i.e., FOCOM = UCOM(NET) + N(0, 1) * SIGCOM(NET)

↓

Compute clock number of receiver; i.e., NTCLK = NET + NTFDC

↓

Update receiver clock; i.e., TC(NTCLK) = TC(NTCLK) + FOCOM

↓

Update ground-to-air communicator clock; i.e., TC(NRTCLK) = TC(NTCLK) + EPSILN

↓

Remove request IR from fire-request list NT; i.e., adjust subscript of each fire-request list entry whose subscript is greater than IR. For example, XFRL(I,NT) → XFRL(I - 1,NT) for I = IR + 1, IR + 2,...,NFR(NT). Then reduce NFR(NT) by one.

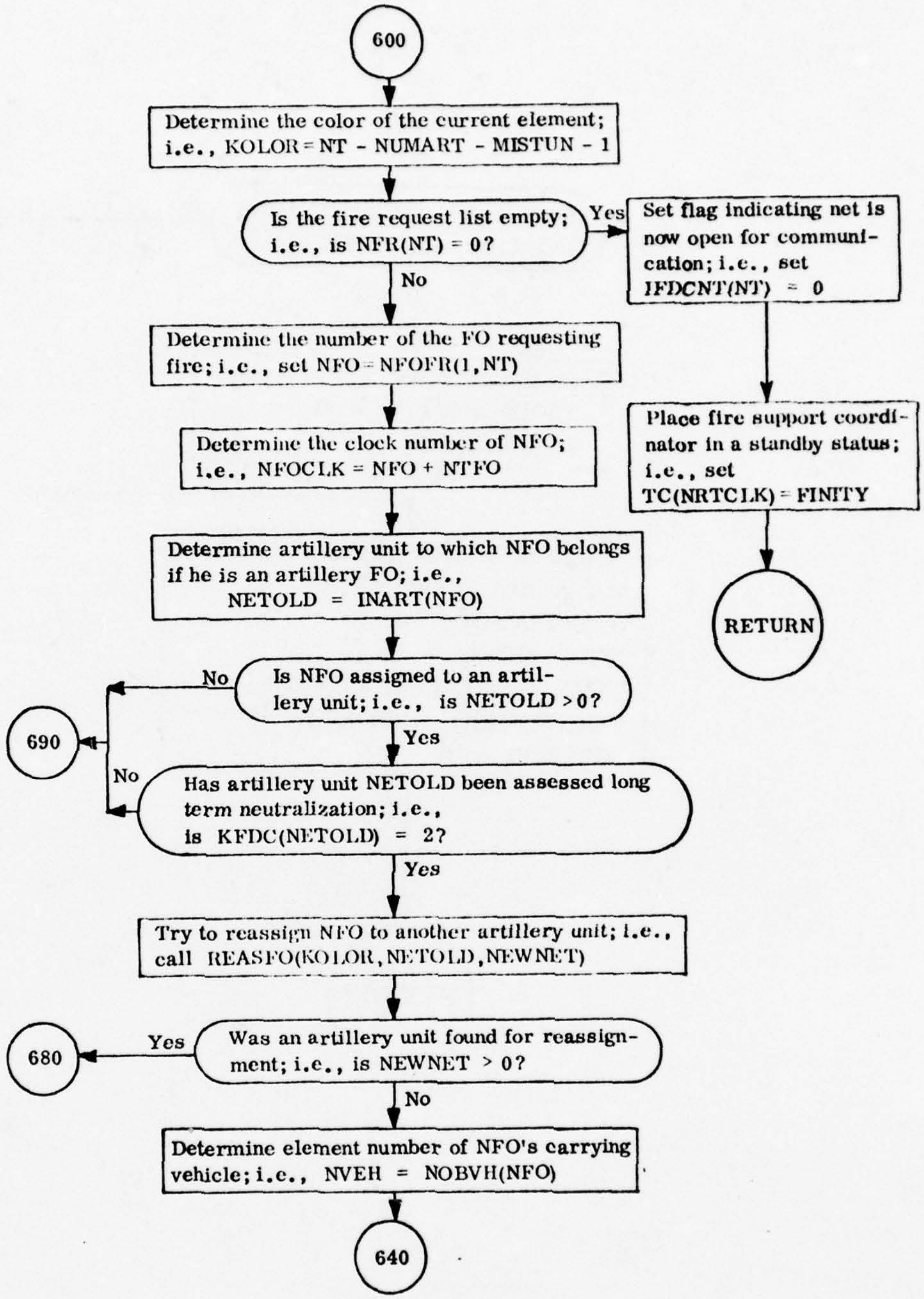
↓

Set flag so that ground-to-air communicator will have a target selection event; i.e., set KFOD(KSUB) = 0

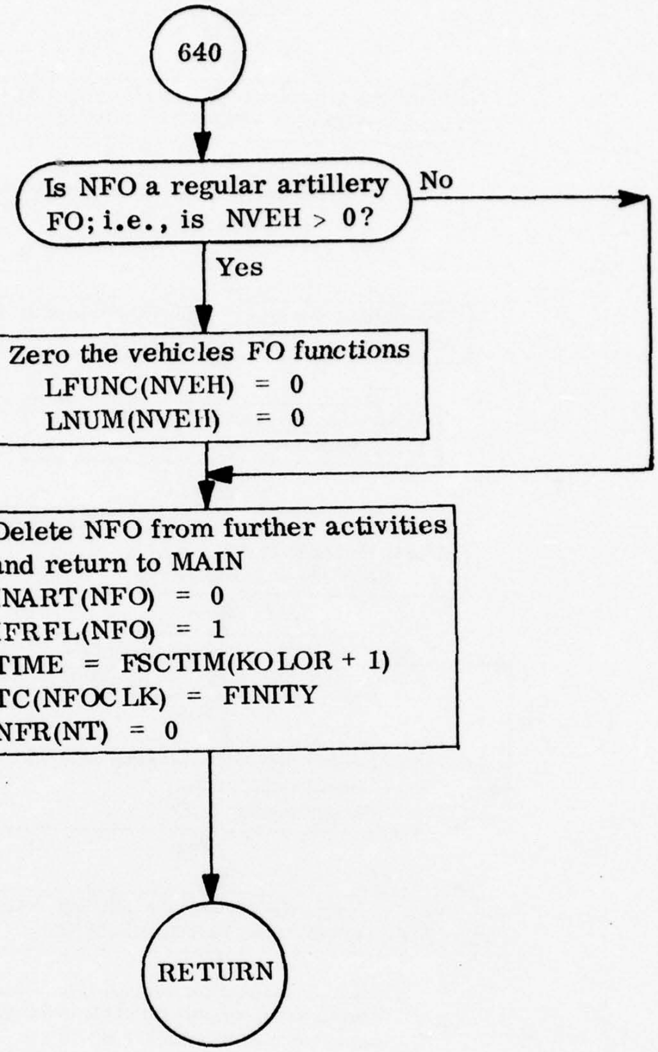
↓

RETURN

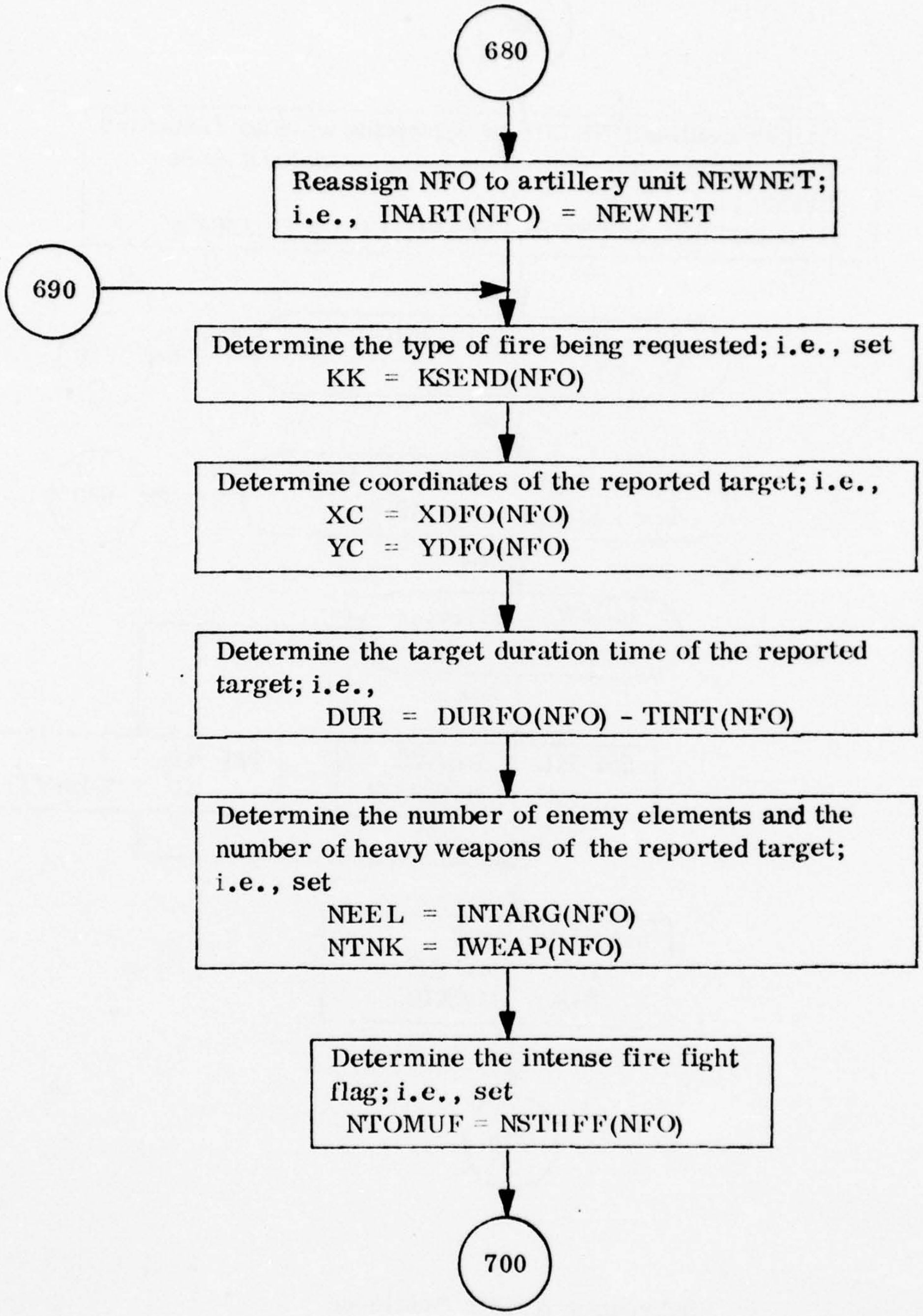
Subroutine AFSC: Continued



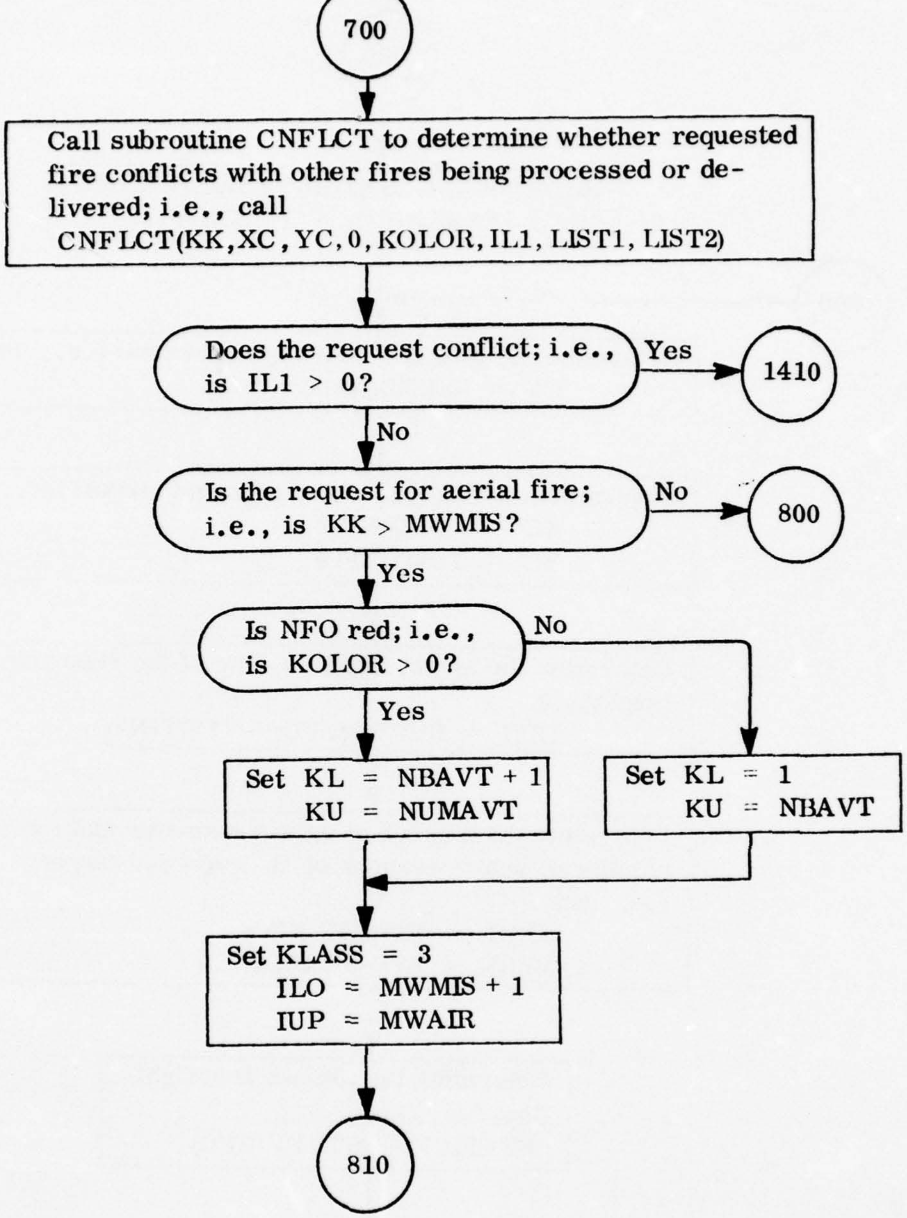
Subroutine AFSC: Continued



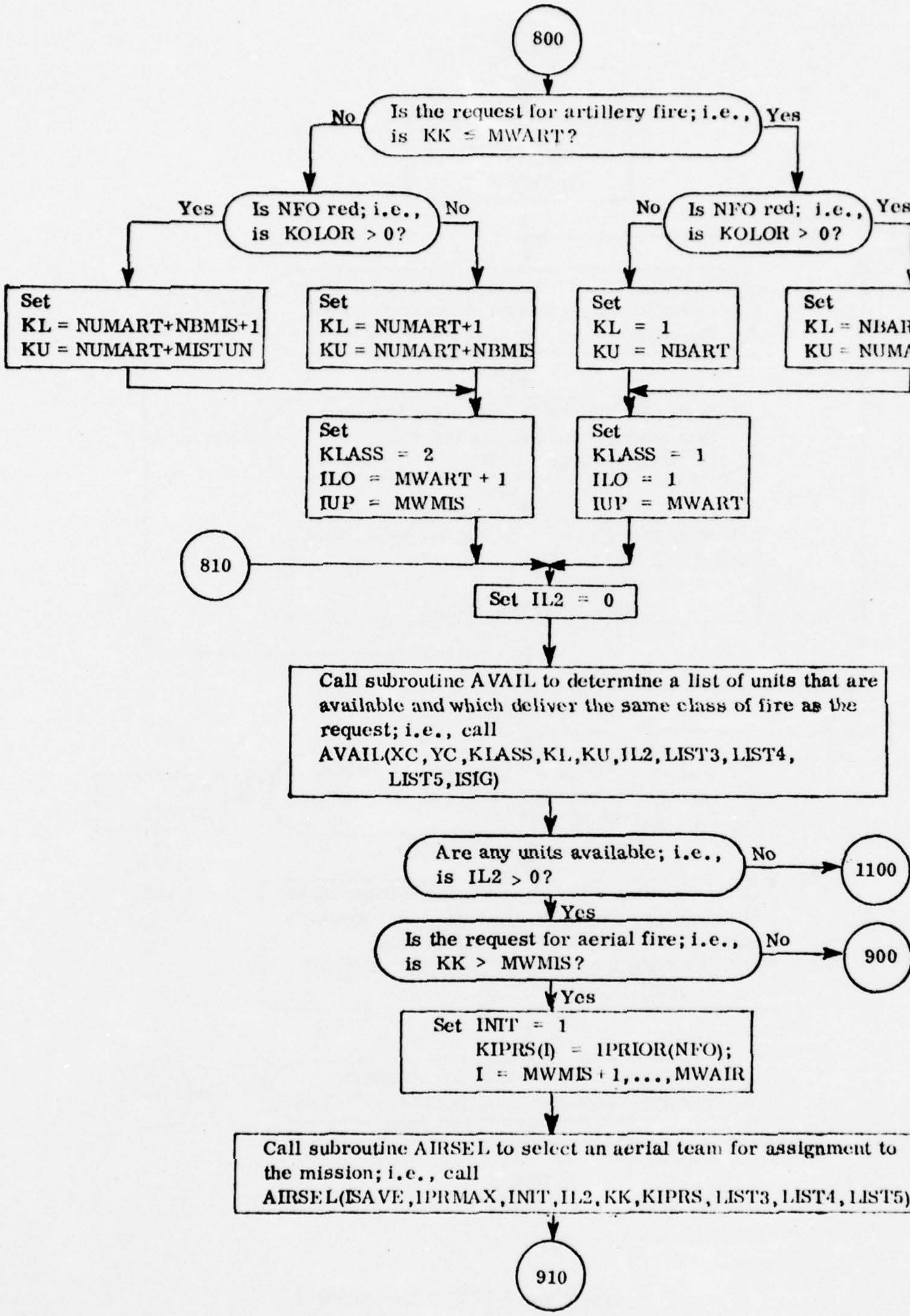
Subroutine AFSC: Continued



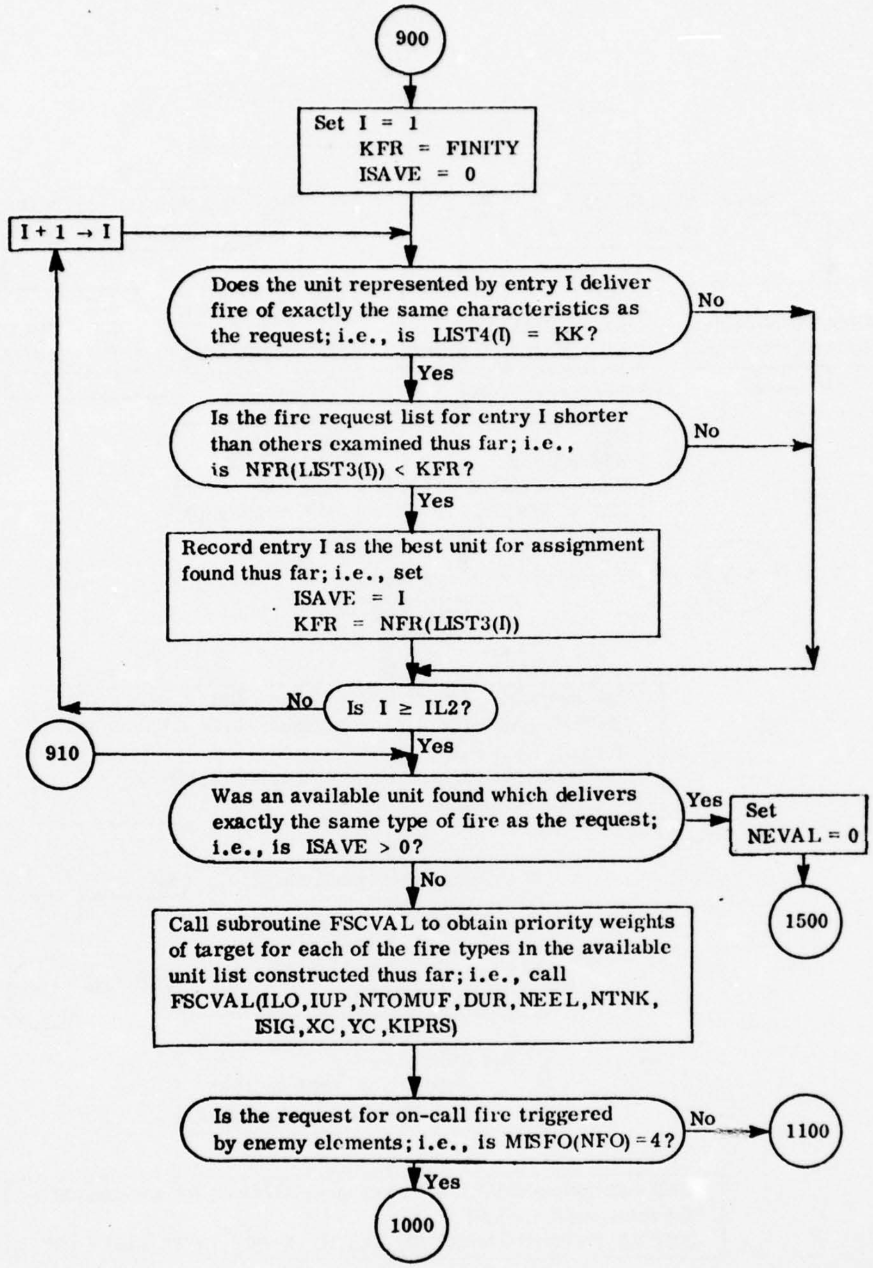
Subroutine AFSC: Continued



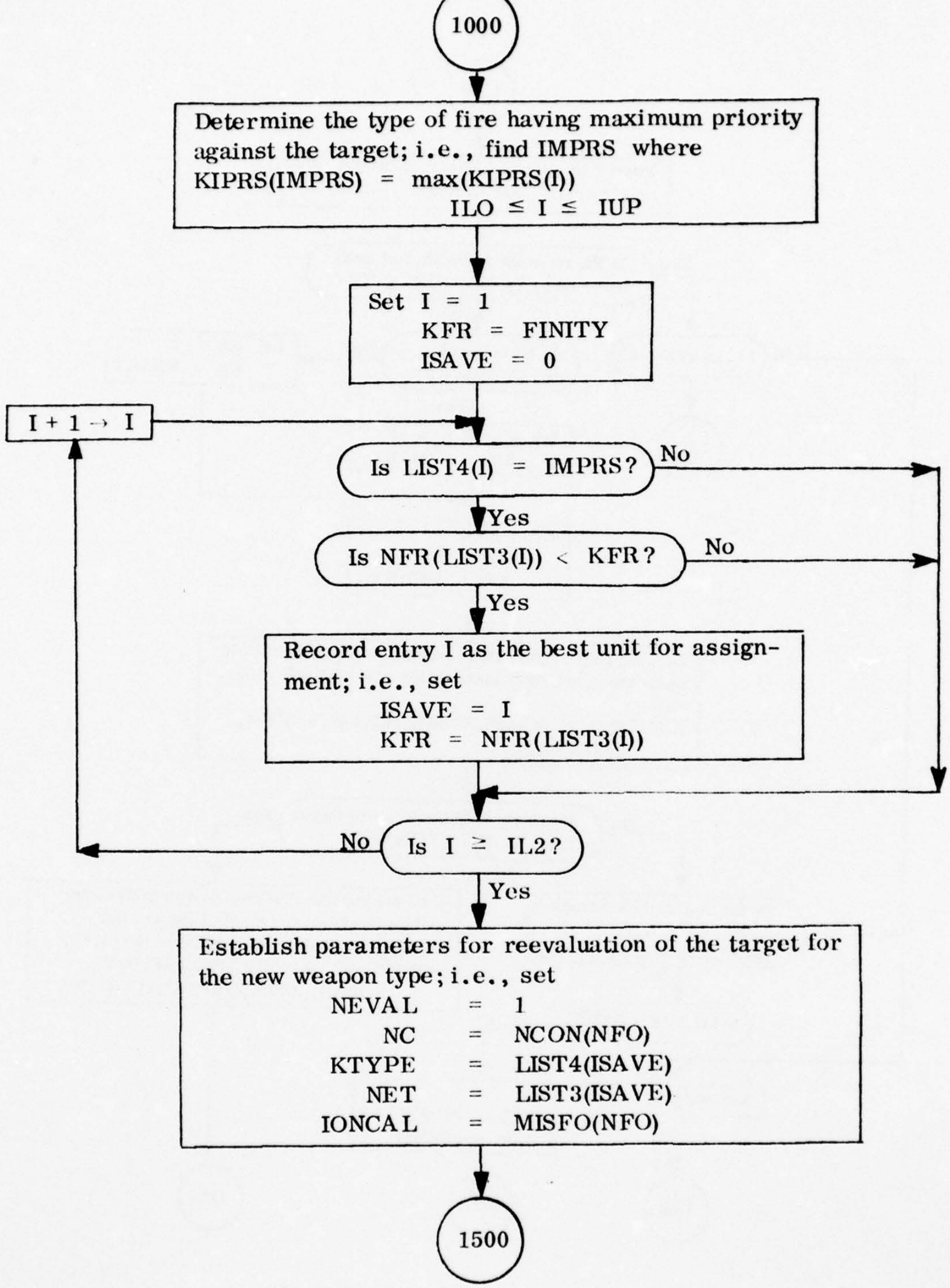
Subroutine AFSC: Continued



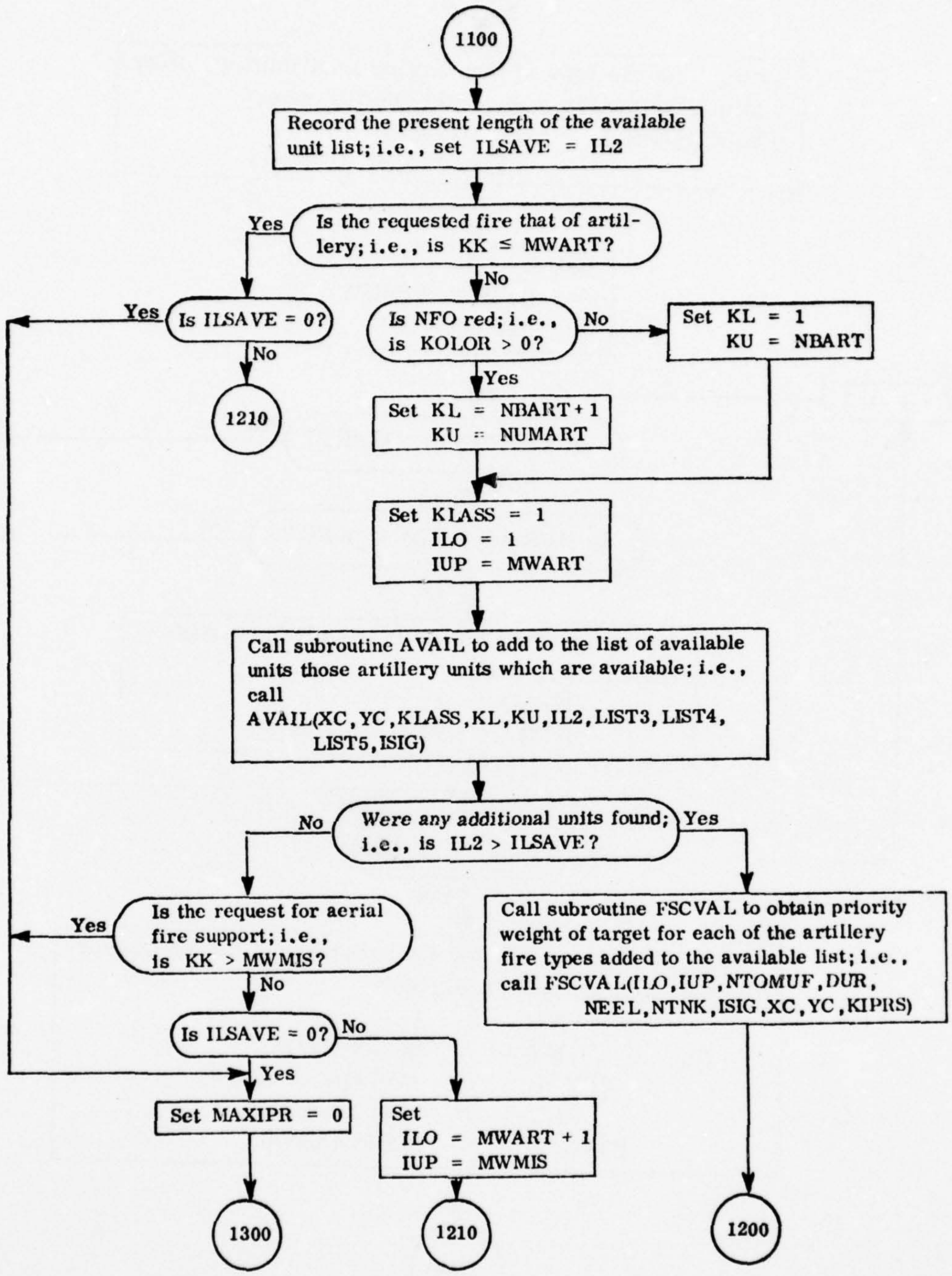
Subroutine AFSC: Continued



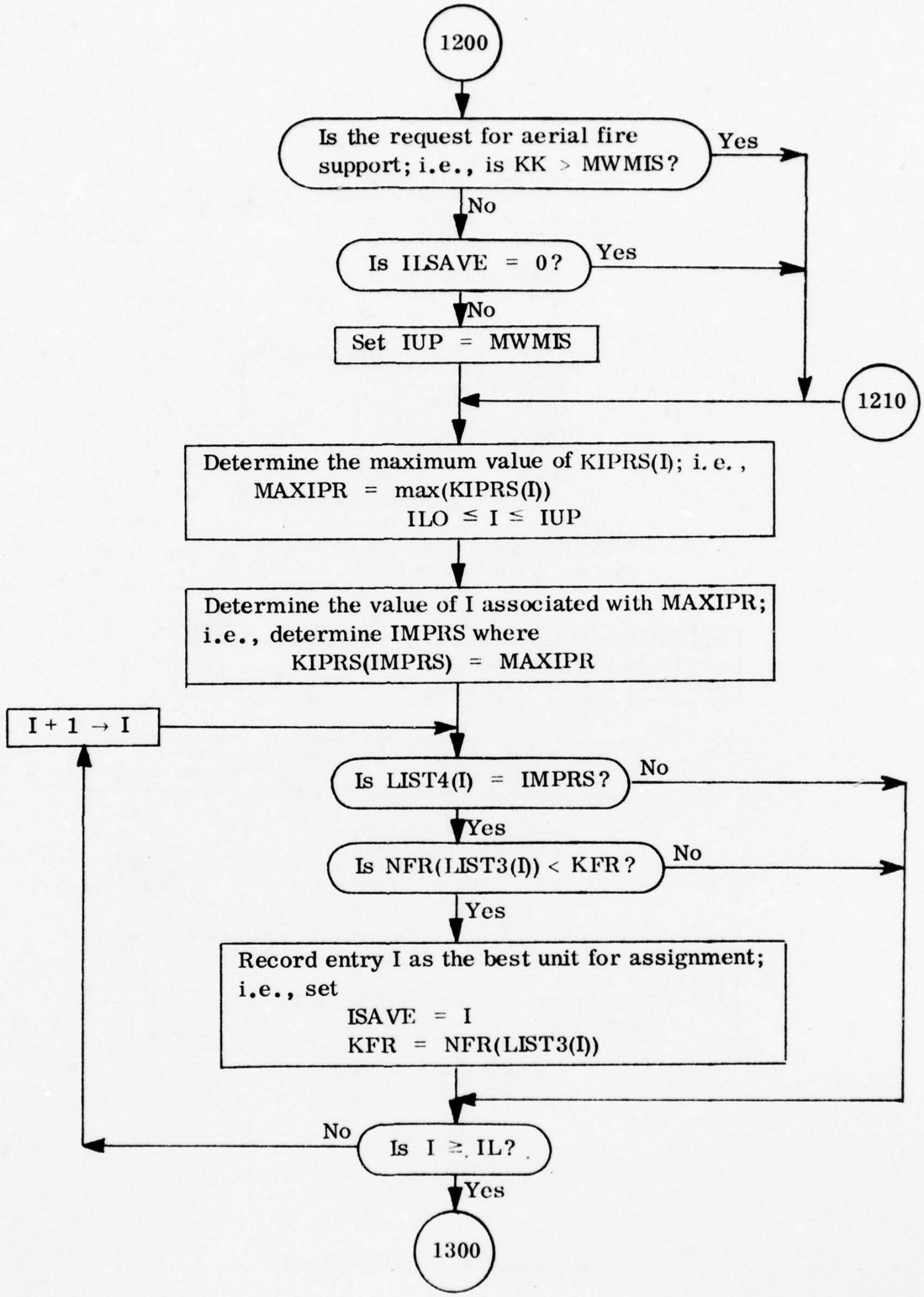
Subroutine AFSC: Continued



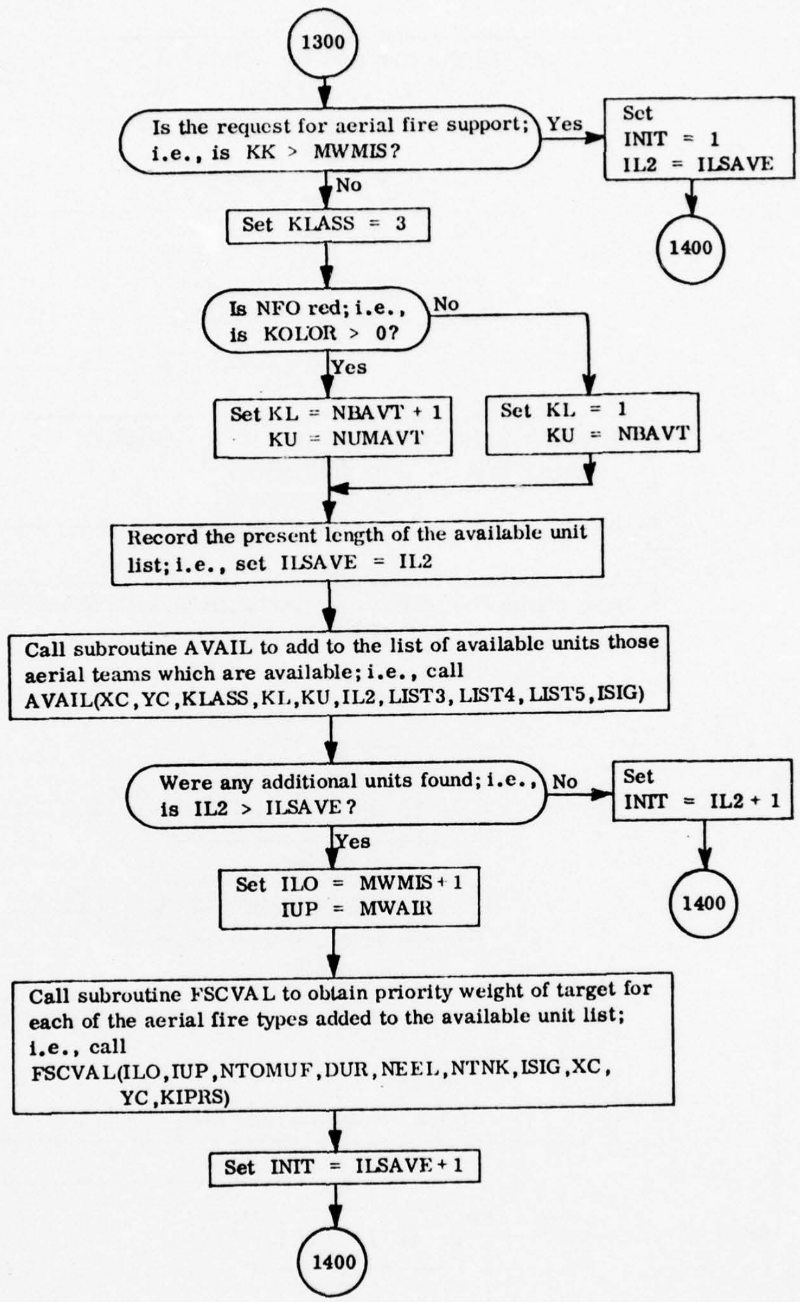
Subroutine AFSC: Continued



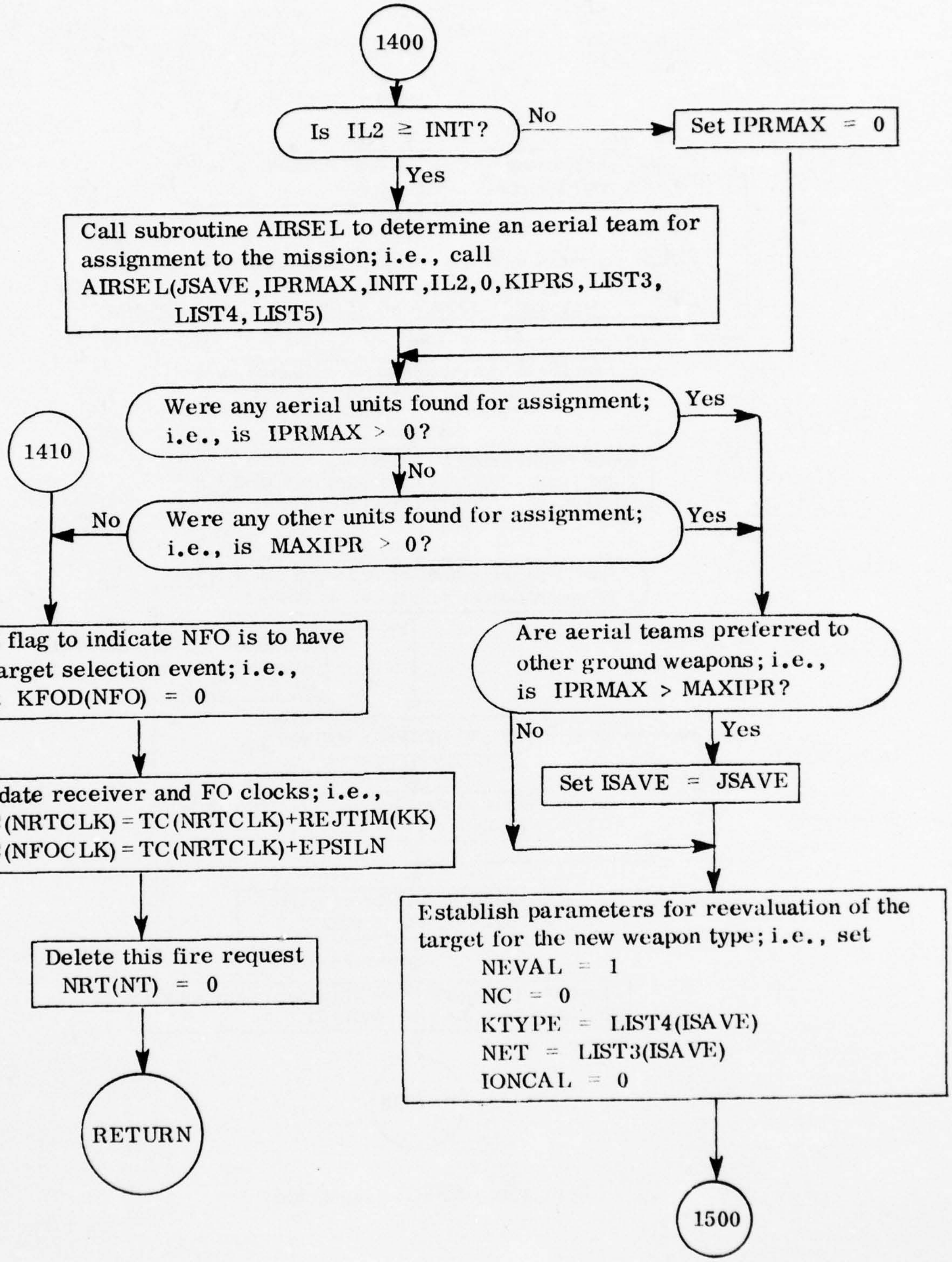
Subroutine AFSC: Continued



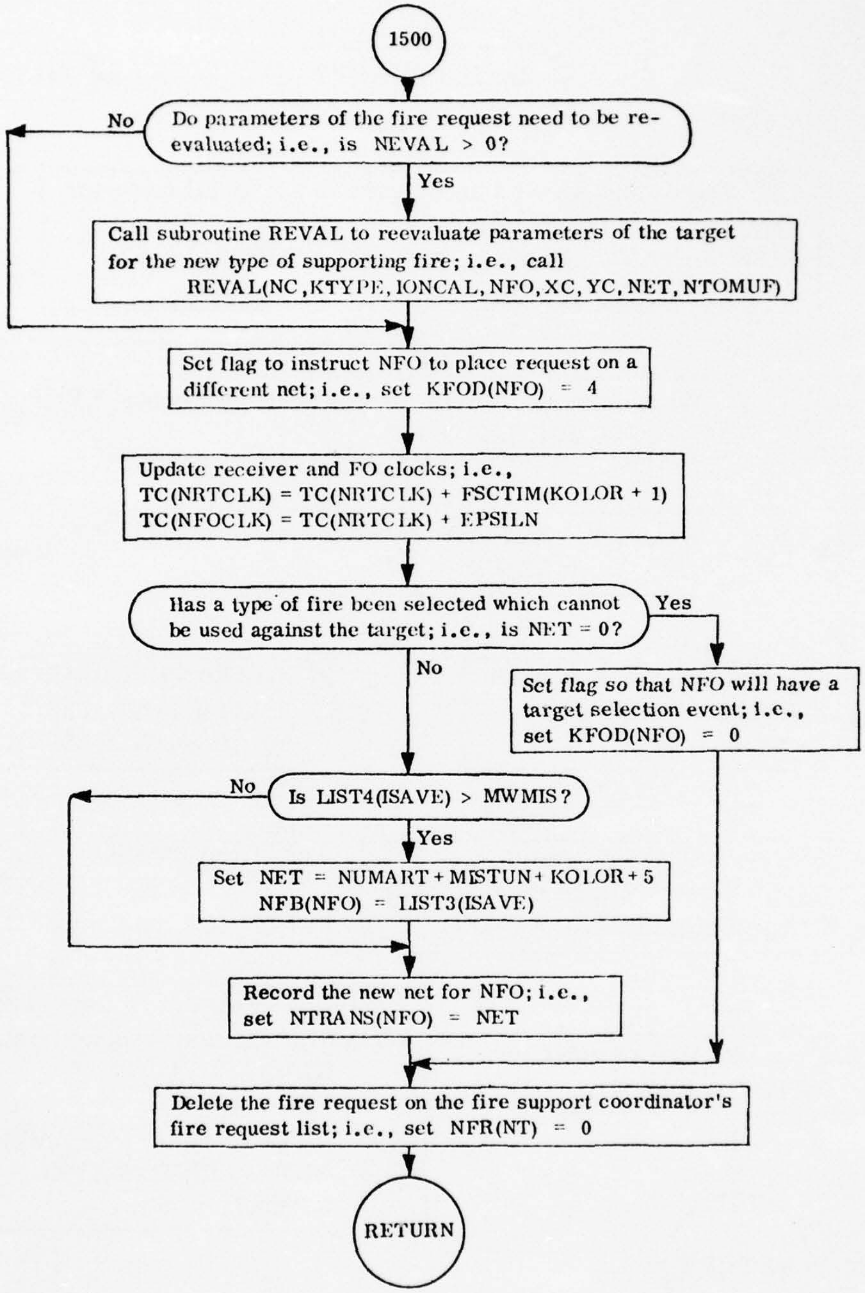
Subroutine AFSC: Continued



Subroutine AFSC: Continued



Subroutine AFSC: Continued



Subroutine AFSC: Continued

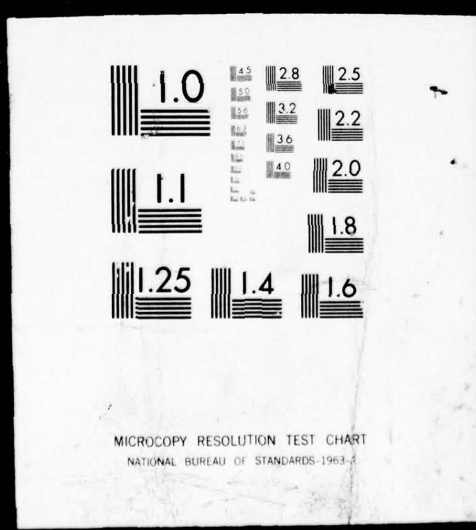
AD-A040 053 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM). VOLUME 2, SECTION--ETC(U)
DEC 71 G M CLARK, R J WILHELM DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-2(U)-SECT-1 NL

4 OF 5
AD
A040053

SIFIED

4 OF 5

AD A040053



Subroutine AIRAMO

PURPOSE: Subroutine AIRAMO determines the total number of destructive and suppressive fire weapons currently available to a specified aerial vehicle section.

CALLING SEQUENCE:

CALL AIRAMO (IASC, KAZ, LAZ, IWC, IAE)

where IASC = aerial vehicle section number,
KAZ = number of point fire weapons currently available,
LAZ = number of suppressive fire weapons currently available,
IWC = aerial vehicle section weapon code, and
IAE = array of element numbers of section IASC.

METHOD: See Chapter 6 of Volume 1.

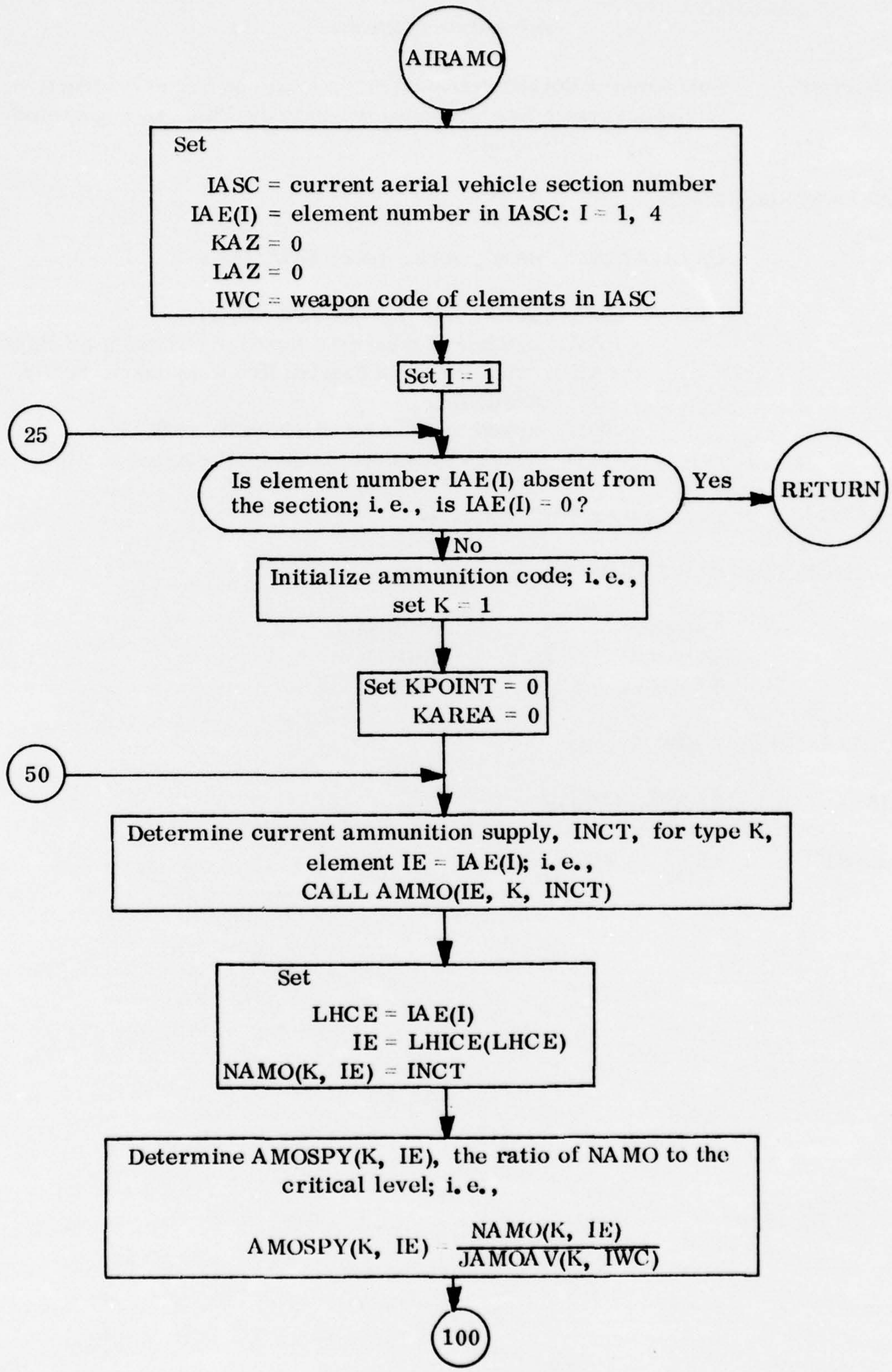
COMMON AREAS REFERENCED:

AMOSPY	KAMOAV
JAMOAV	LHICE
KAMMAX	NAMO

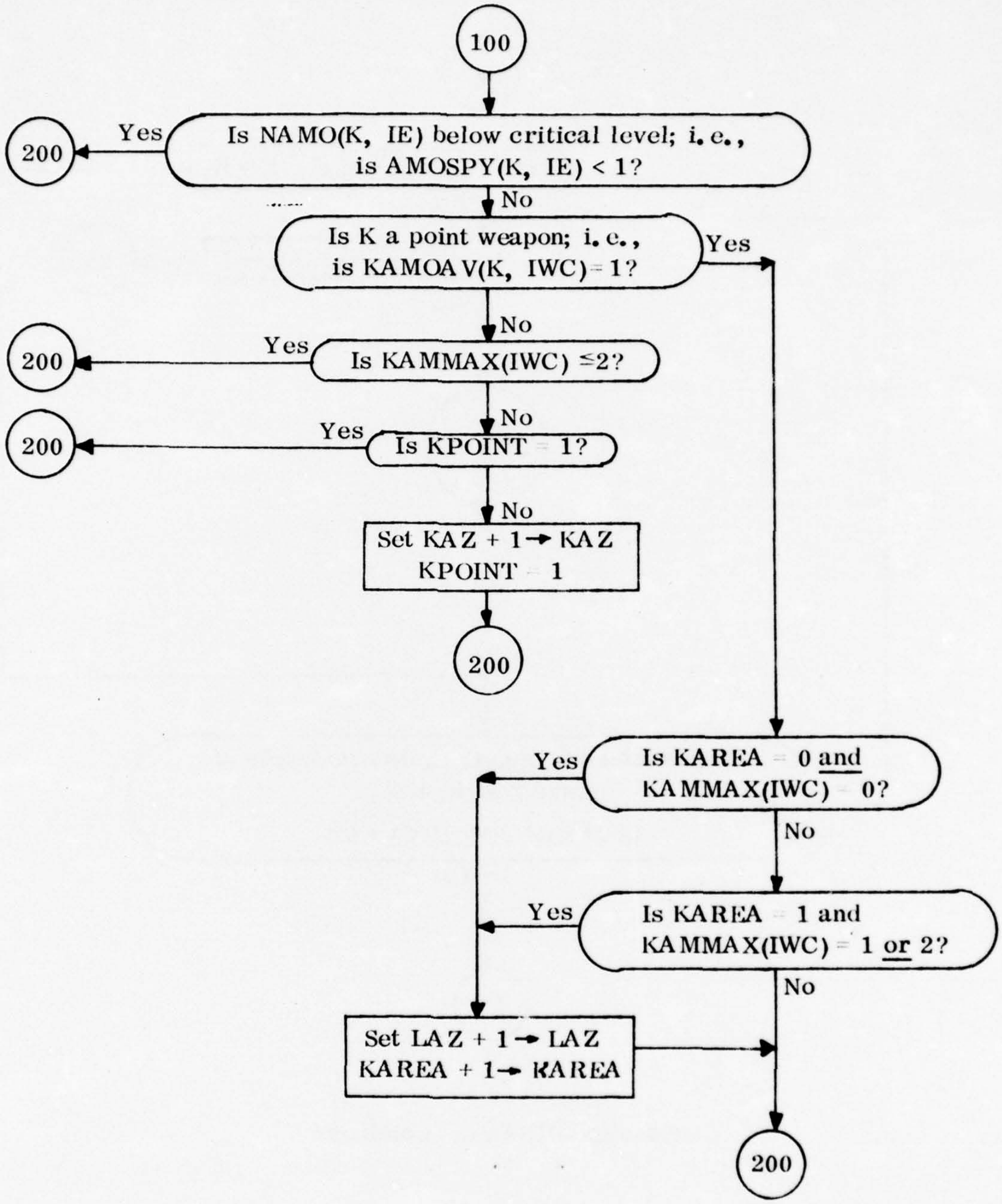
SUBROUTINES USED: AMMO

AIRAMO CALLED BY: AIRFIR

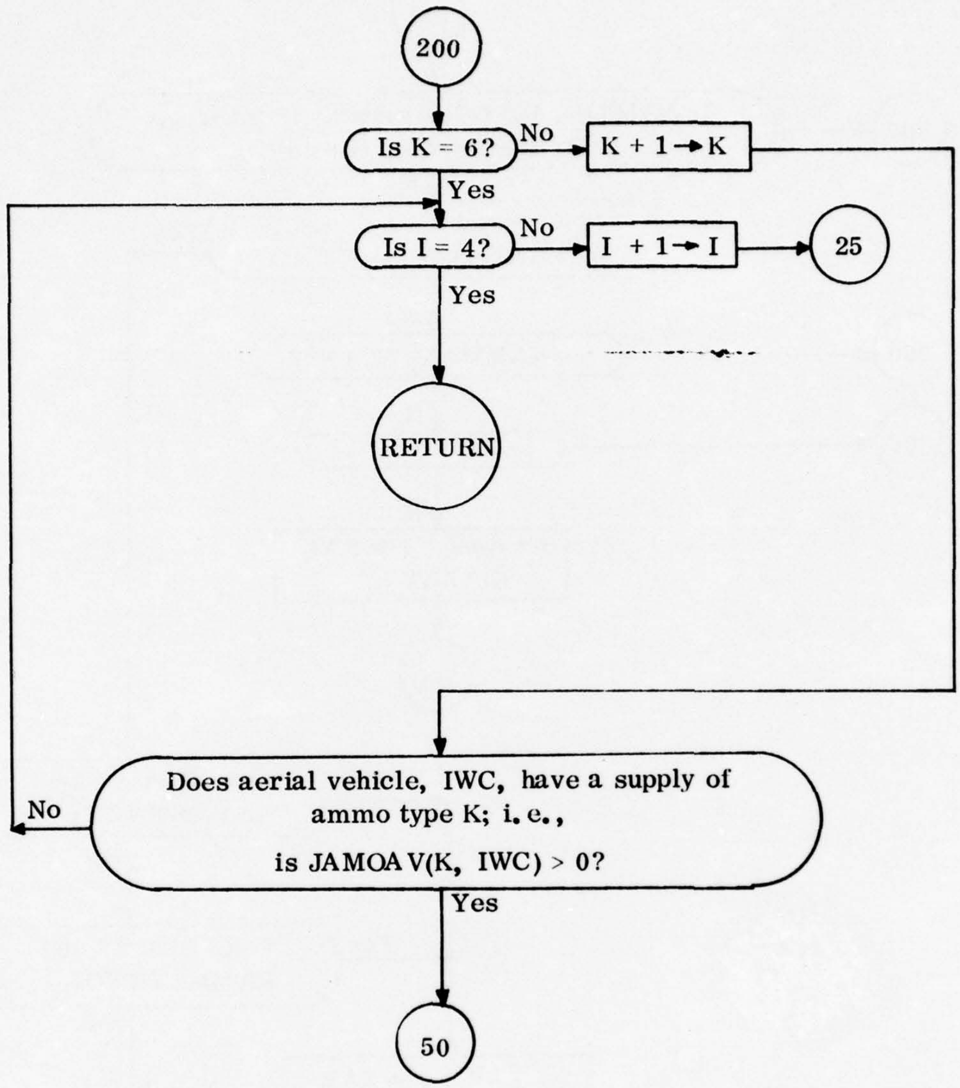
LENGTH: $550_{16} = 1360_{10}$ bytes



Subroutine AIRAMO



Subroutine AIRAMO: Continued



Subroutine AIRAMO: Continued

Subroutine AIRFB

PURPOSE: Subroutine AIRFB simulates the decisions and resulting communications activities required for initiation and termination of missions by an aerial unit.

CALLING SEQUENCE:

CALL AIRFB(NAT,TIME)

where

NAT = aerial unit number

TIME = event time

METHOD: See Chapter 4 of Volume 1.

COMMON AREAS REFERENCED:

DURFB	KANCEL	MESARR	SEQPAR
DURRL	KFO	MESCON	SIGSEN
ECLOCK	KFOD	MISFRL	STRMIS
ENDMIS	KMANU	MISTYP	STRTIM
EVTIM	KPRIOR	MPTR	TINIT
ICECOM	KPTFRL	NFB	USEN
IFBMIS	LARFB	NFOFR	WAITAD
IFDCNT	LMUFL	NFR	WAITFO
IFRFL	LSTFDC	NRNDFR	XD
IPHASE	LWCOD	NTELE	XFRL
IPRIRR	MAINPR	NUMBER	YD
ITMASS	MANACT	NVOLM	YFRL
IUNACT	MANLDR	NXFRL	

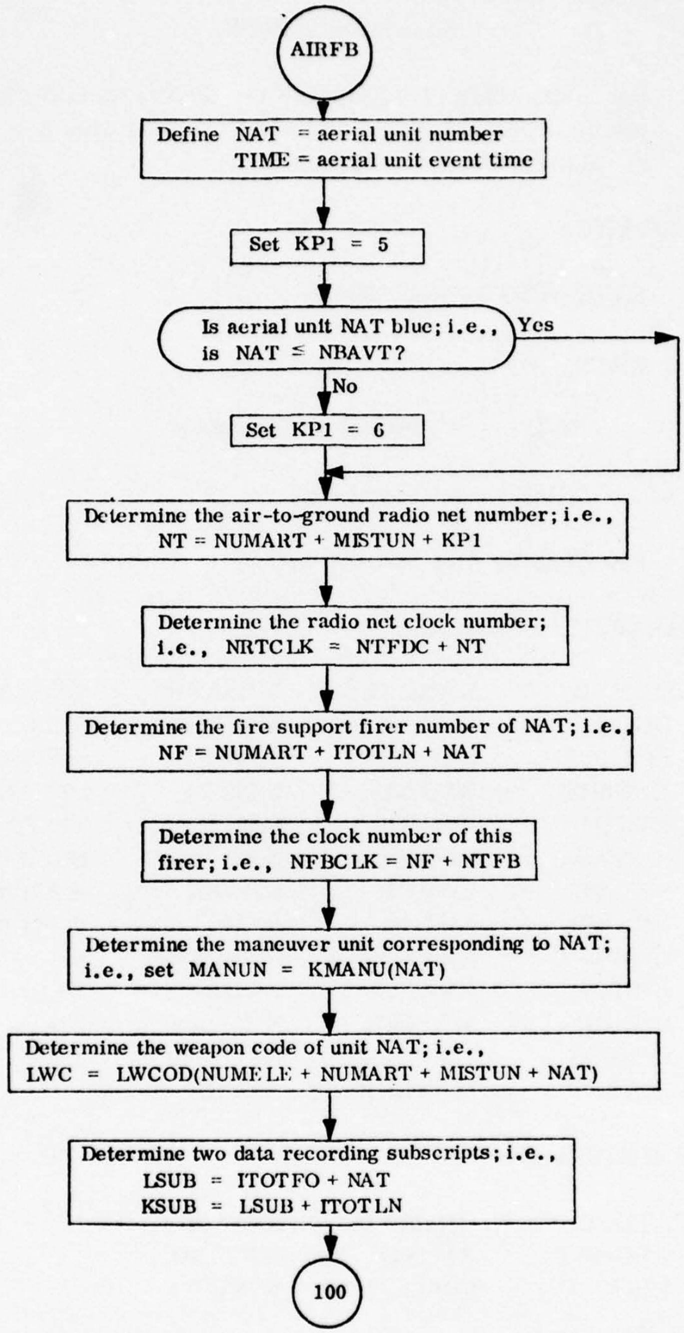
SUBROUTINES REQUIRED:

ATDEC	PRMSET	STAXIS
CBCONT	RANND	WRTFRL
ELOC	STACLK	WRITFO

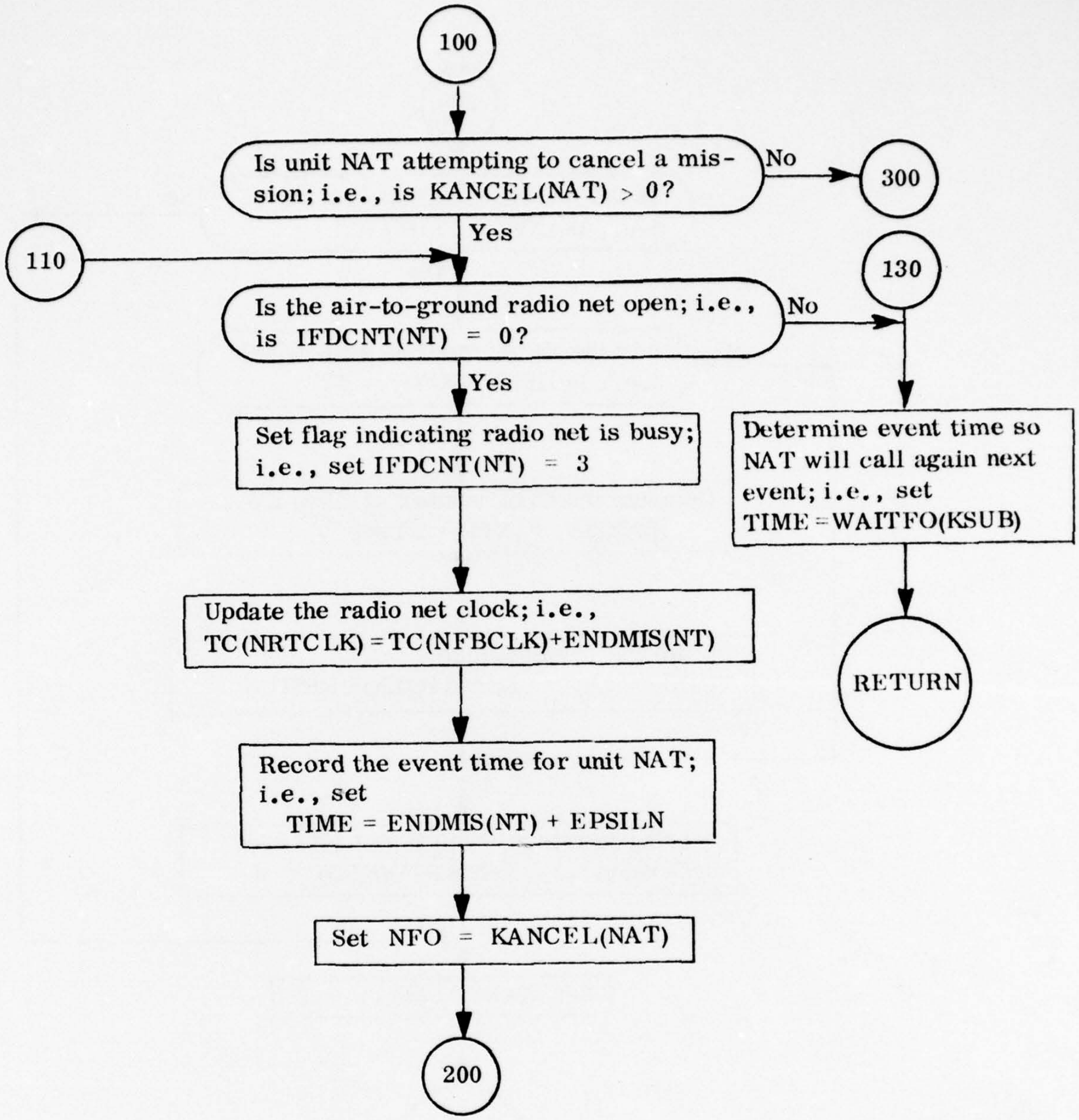
AIRFB CALLED BY:

MAIN PROGRAM

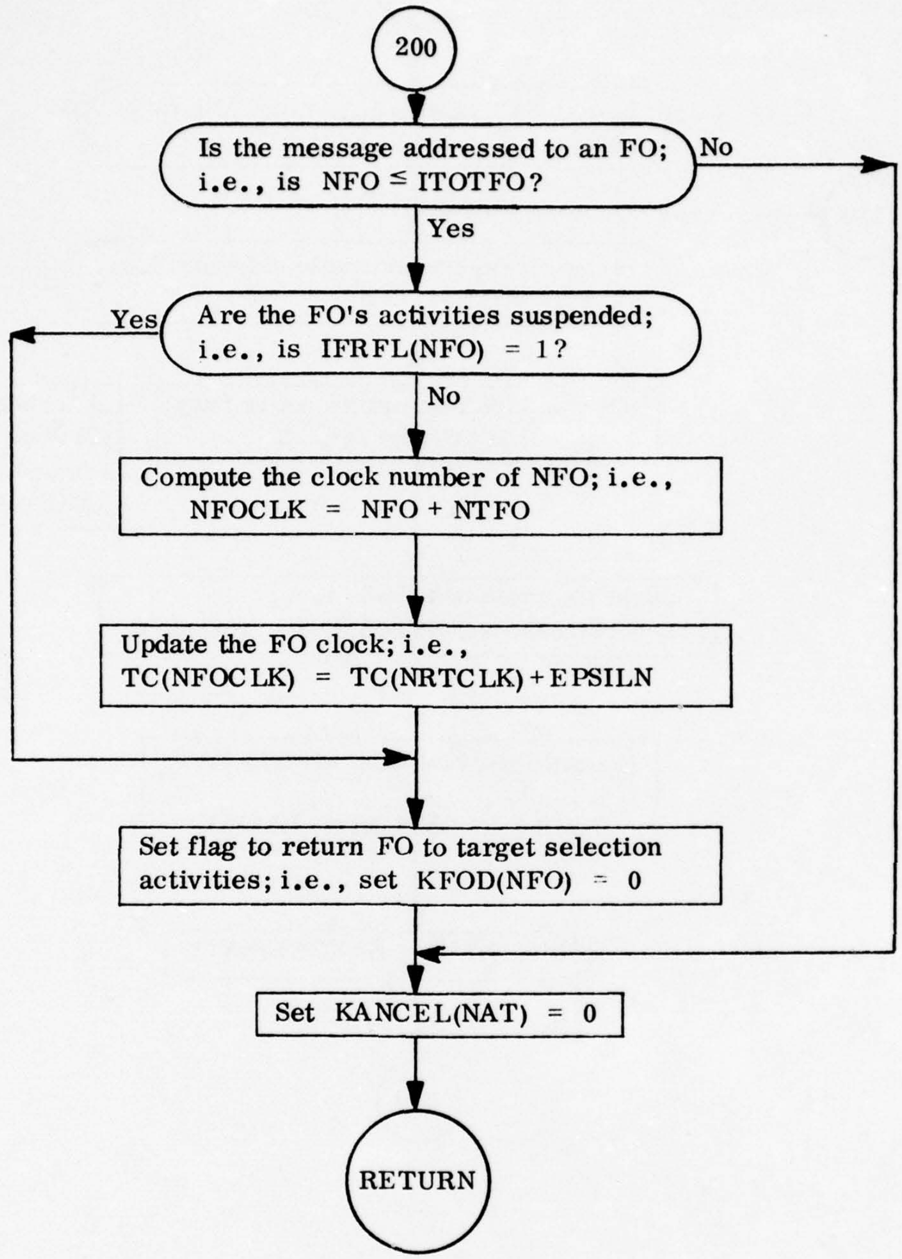
LENGTH: $3B9A_{16}$ 15258_{10} bytes



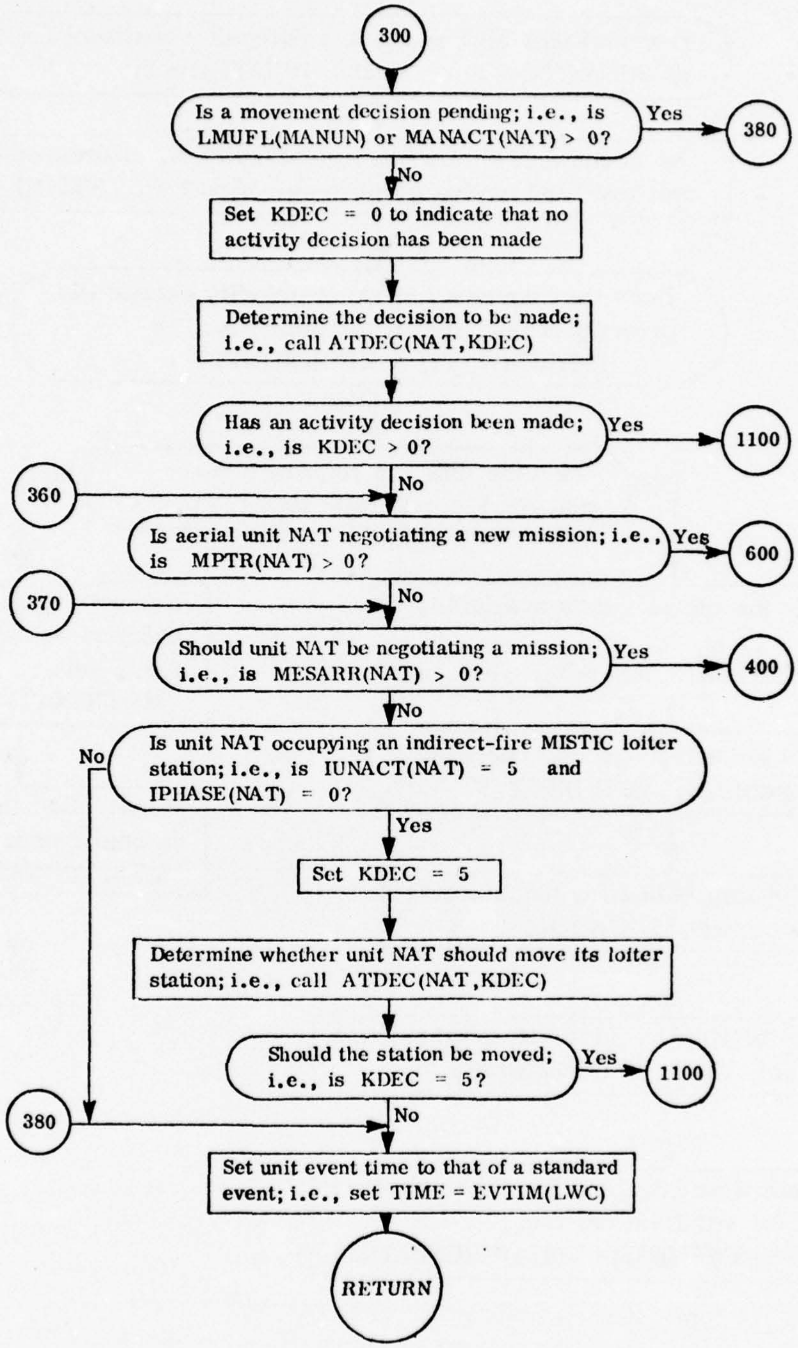
Subroutine AIRFB: Aerial Unit Mission Assignment



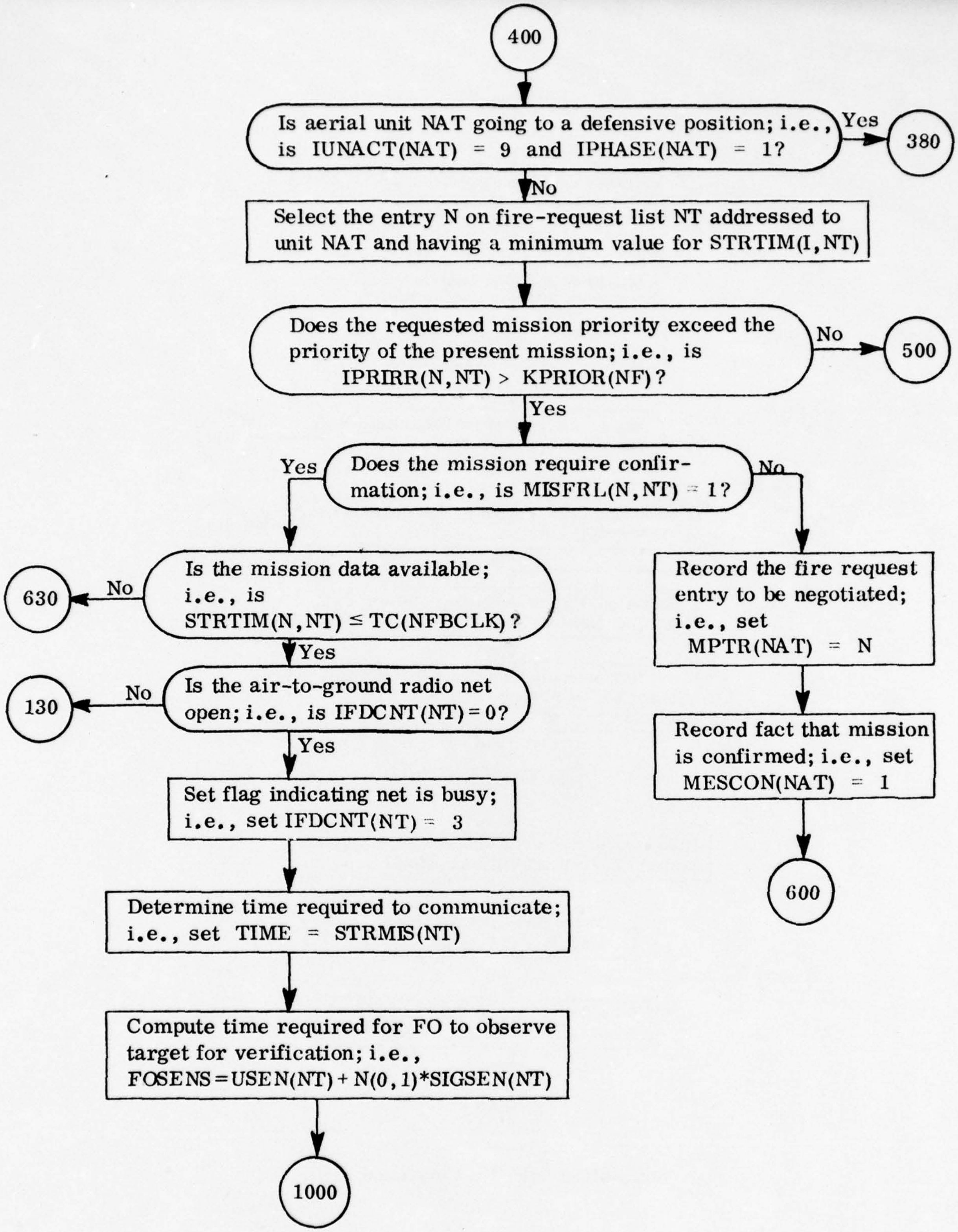
Subroutine AIRFB: Continued



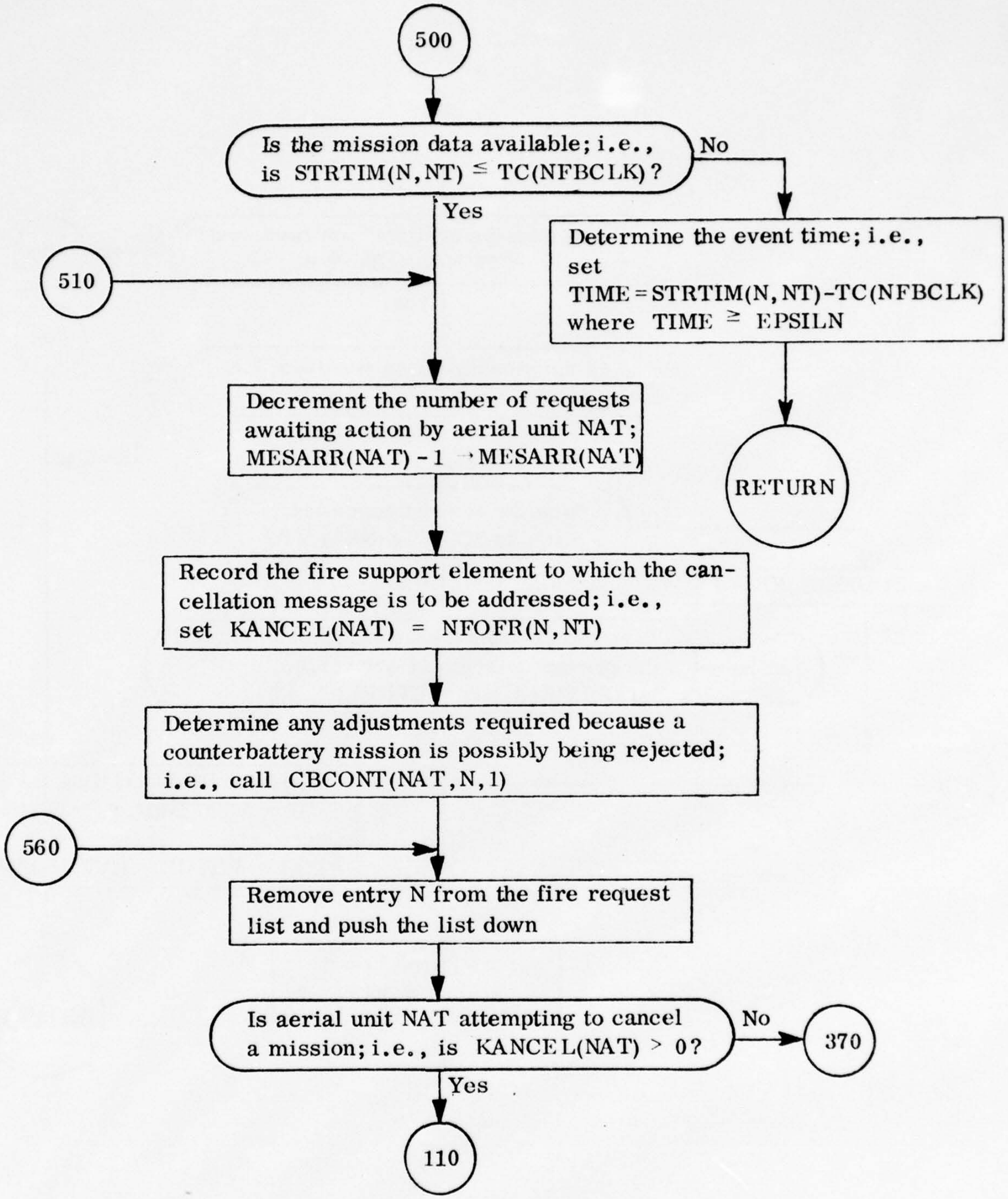
Subroutine AIRFB: Continued



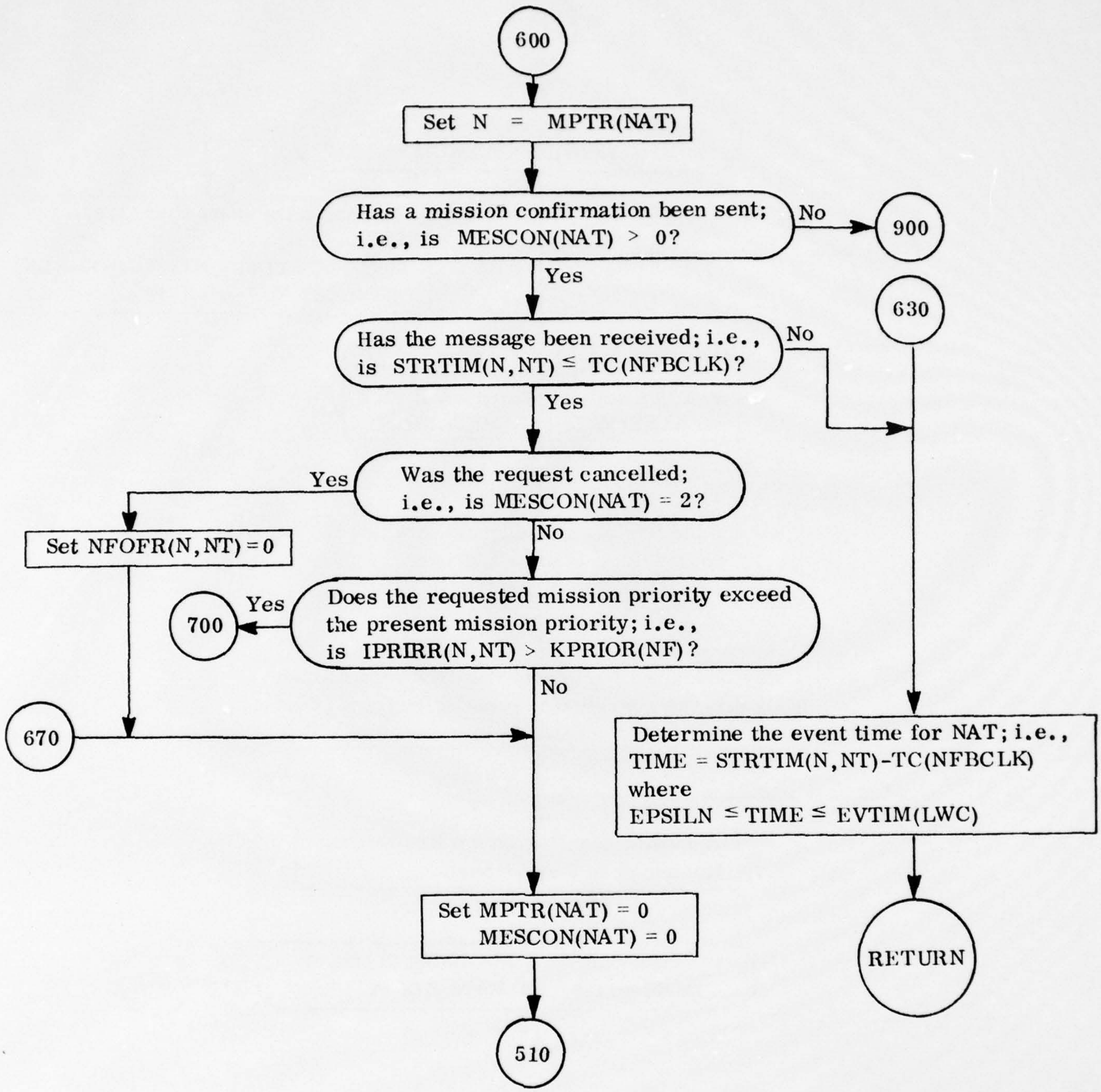
Subroutine AIRFB: Continued



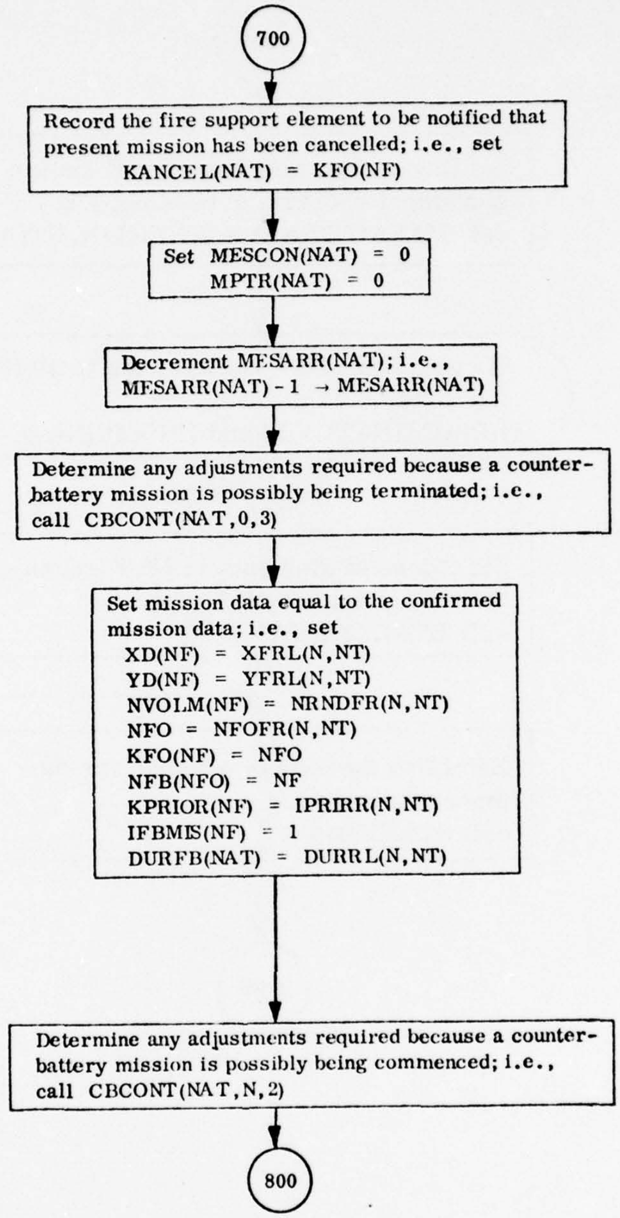
Subroutine AIRFB: Continued



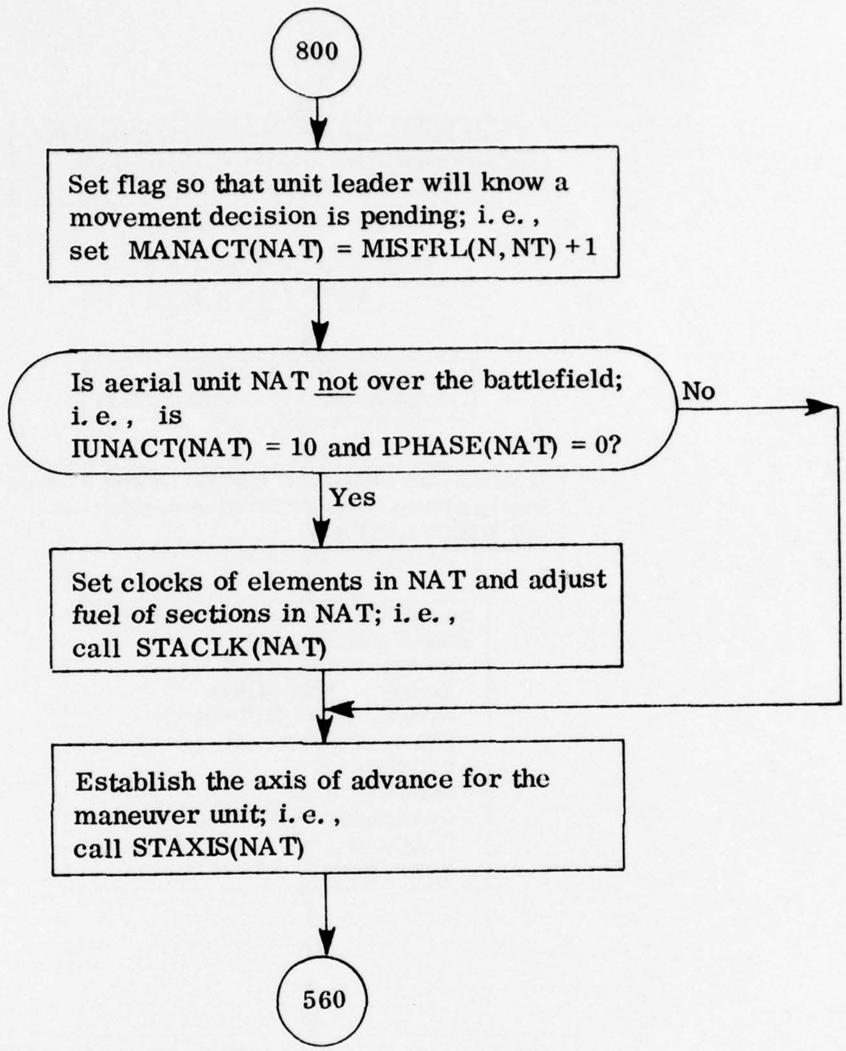
Subroutine AIRFB: Continued



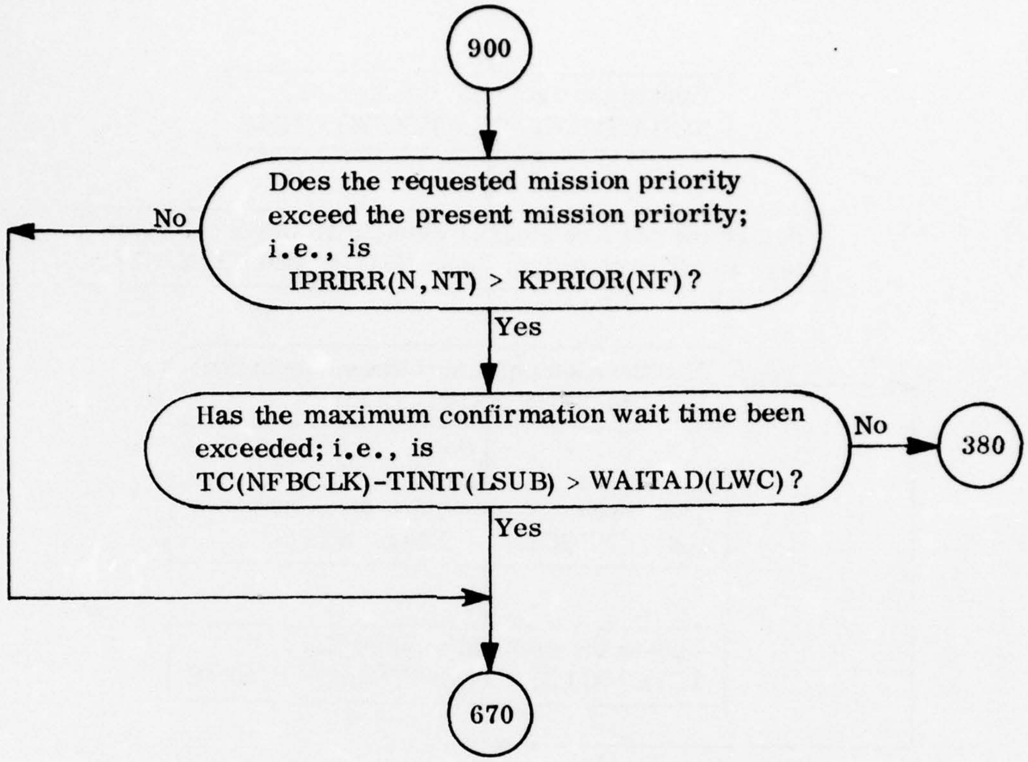
Subroutine AIRFB: Continued



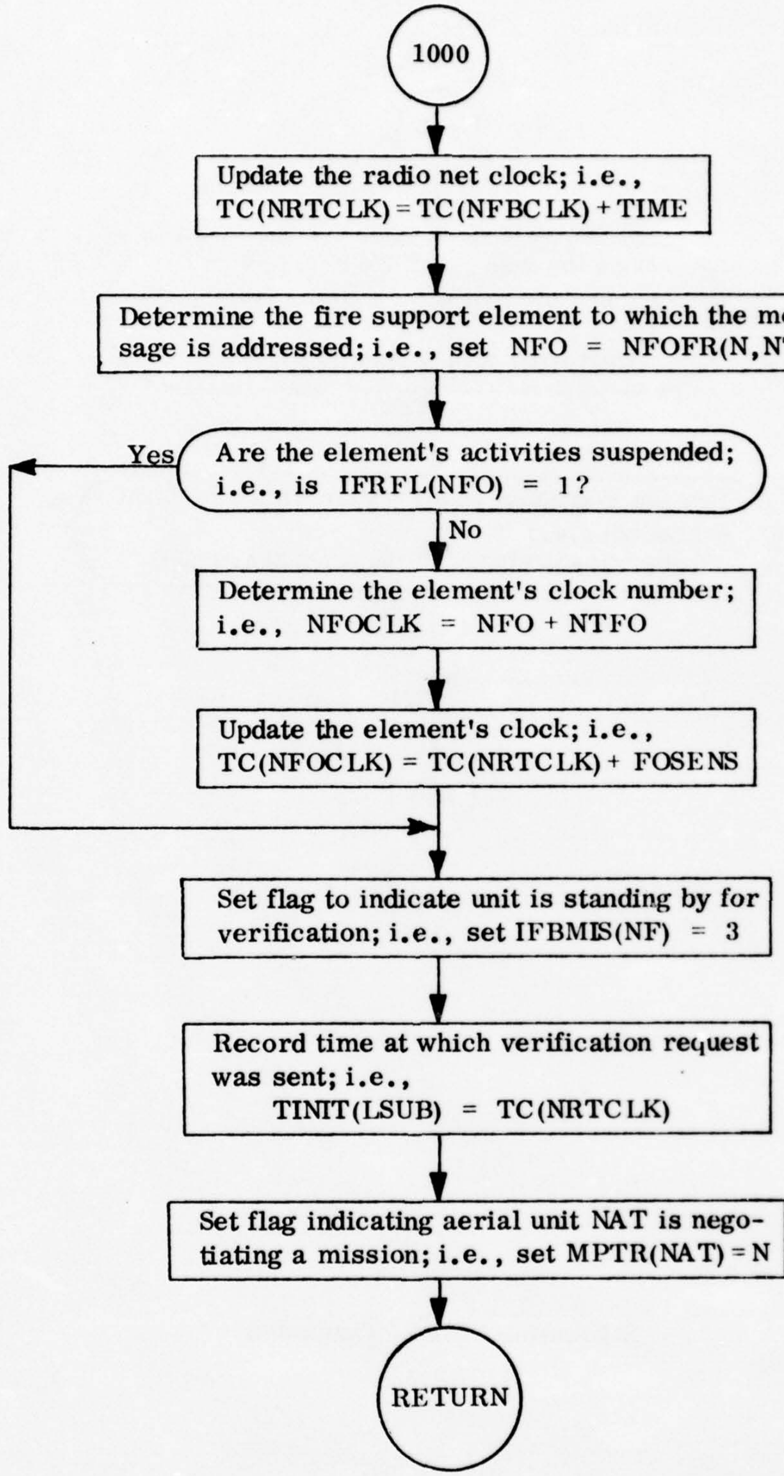
Subroutine AIRFB: Continued



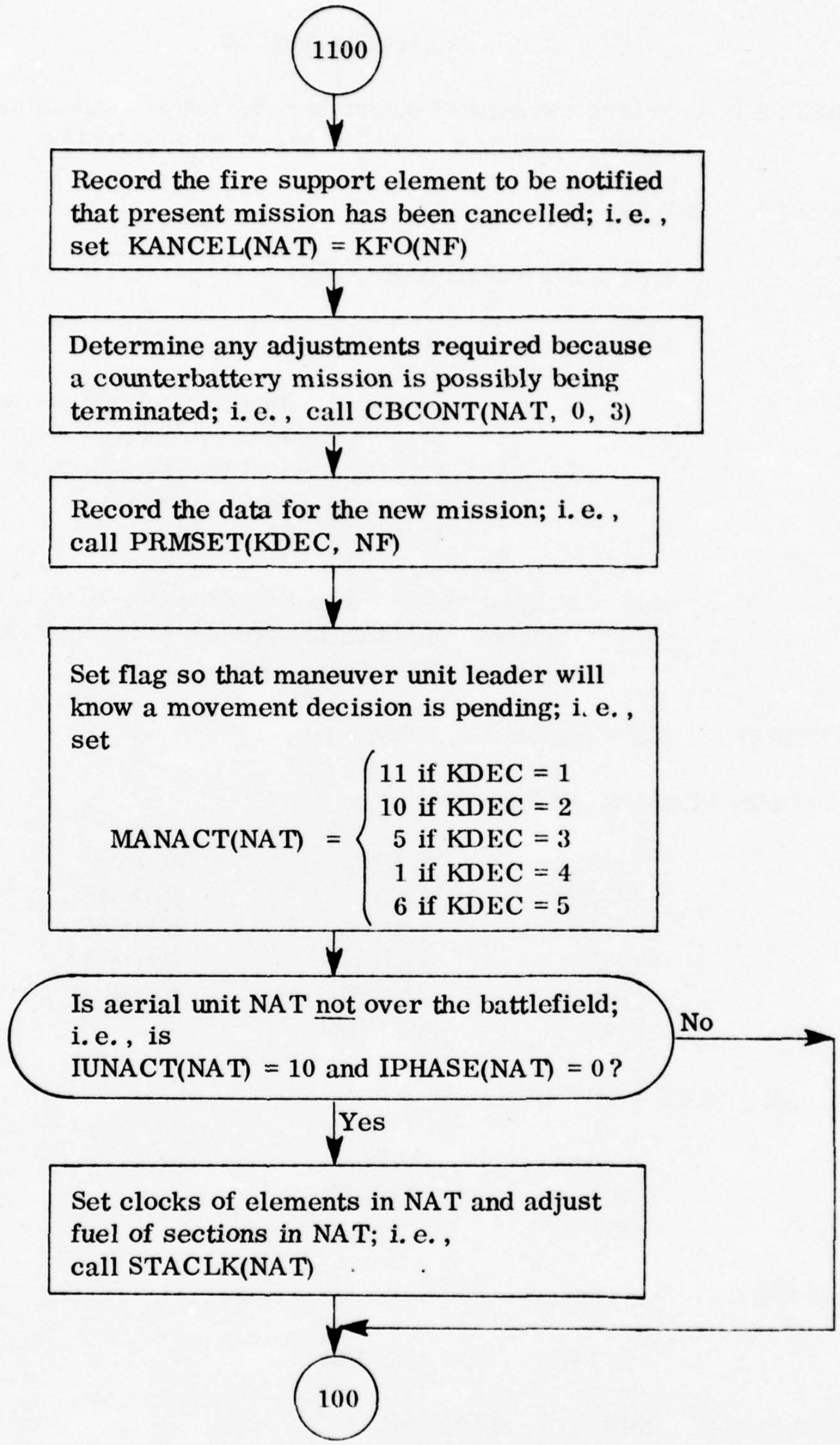
Subroutine AIRFB: Continued



Subroutine AIRFB: Continued



Subroutine AIRFB: Continued



Subroutine AIRFB: Continued

Subroutine AIRFIR

PURPOSE: Subroutine AIRFIR determines the targets and ammunitions for engagement for a specified aerial vehicle section.

CALLING SEQUENCE:

CALL AIRFIR (ITGASN, JJJ)

where

ITGASN = {
- 0 no target selection made due to undesirable complex availability
- 1 no target selection made due to no knowledge of enemy ground elements
- 2 target selection made

JJJ = flag passed to subroutine AIRPOR; a value of 1 denotes priority weighting factors due to engagement by friendly air elements are to be ignored.

METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

BLUDET	ITYPA	NOBVH
DESAIR	KFO	NUMBER
ELOCX	LHICE	RADINC
ELOCY	LKILL	RADMAX
ICECOM	LWCOD	REDDET
ISORG	MANHEL	

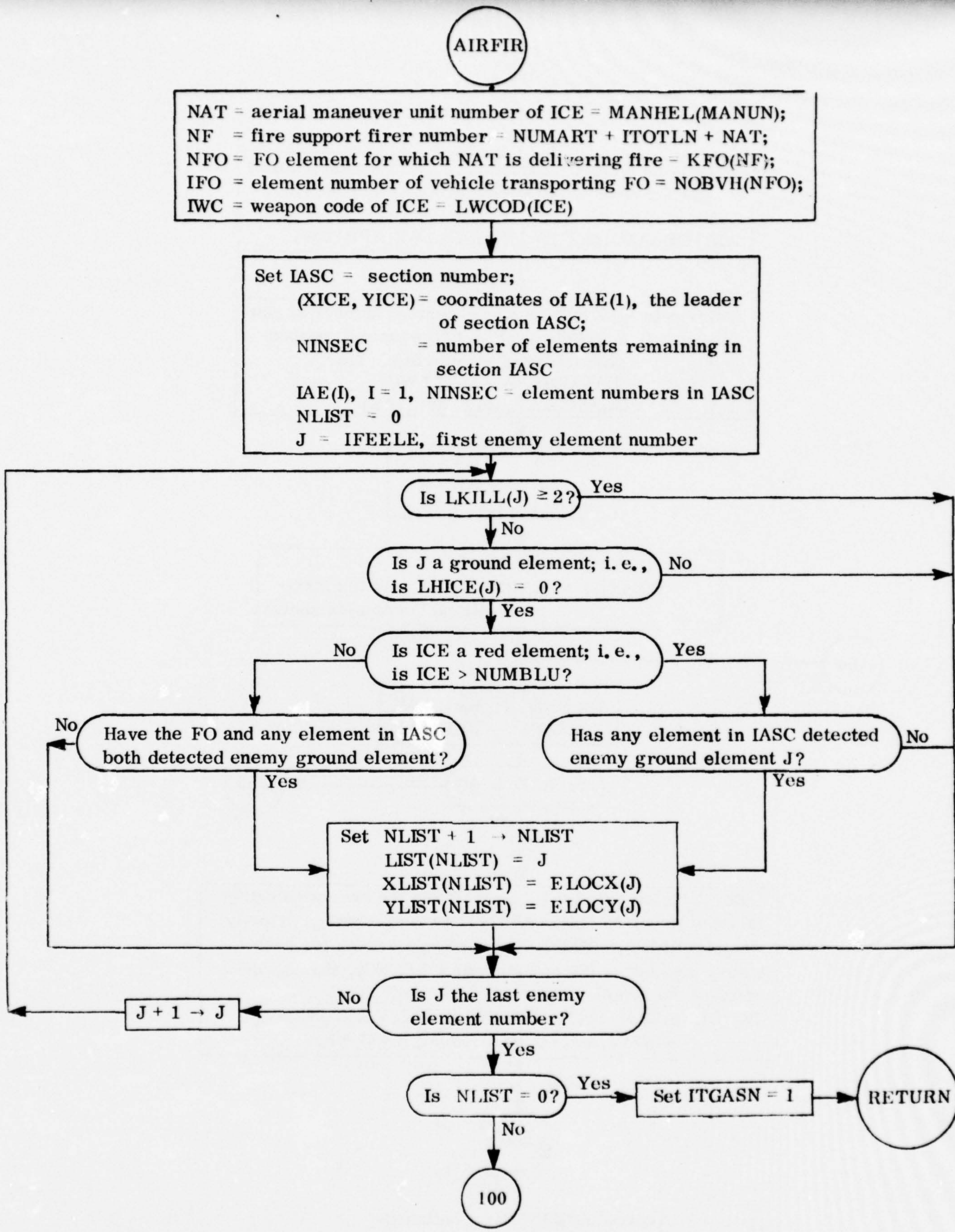
SUBROUTINES REQUIRED:

AIRPOR	TARCEN
AIRAMO	WASAIR
CONVRT	

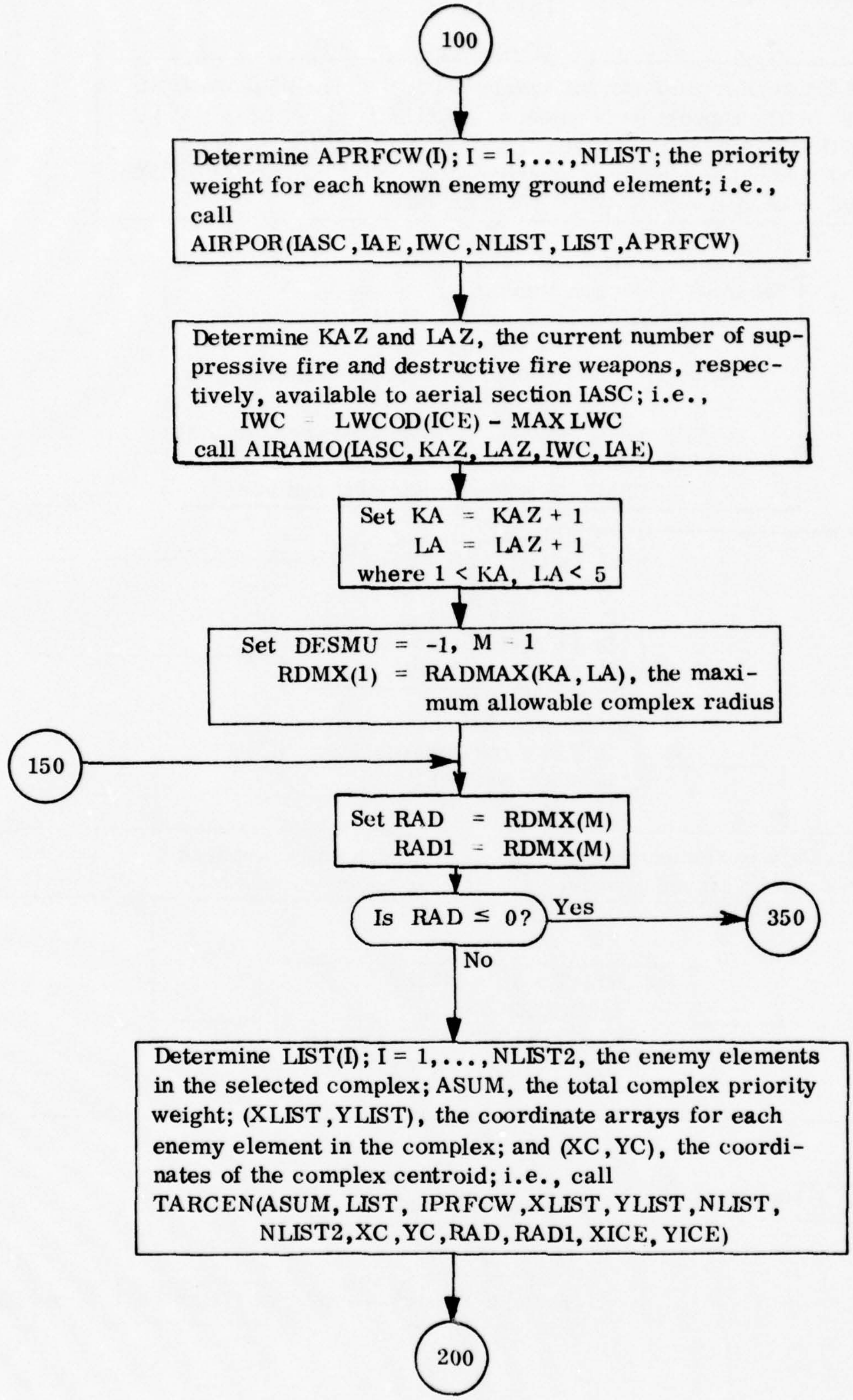
AIRFIR CALLED BY:

ATKPRM	HELFIR

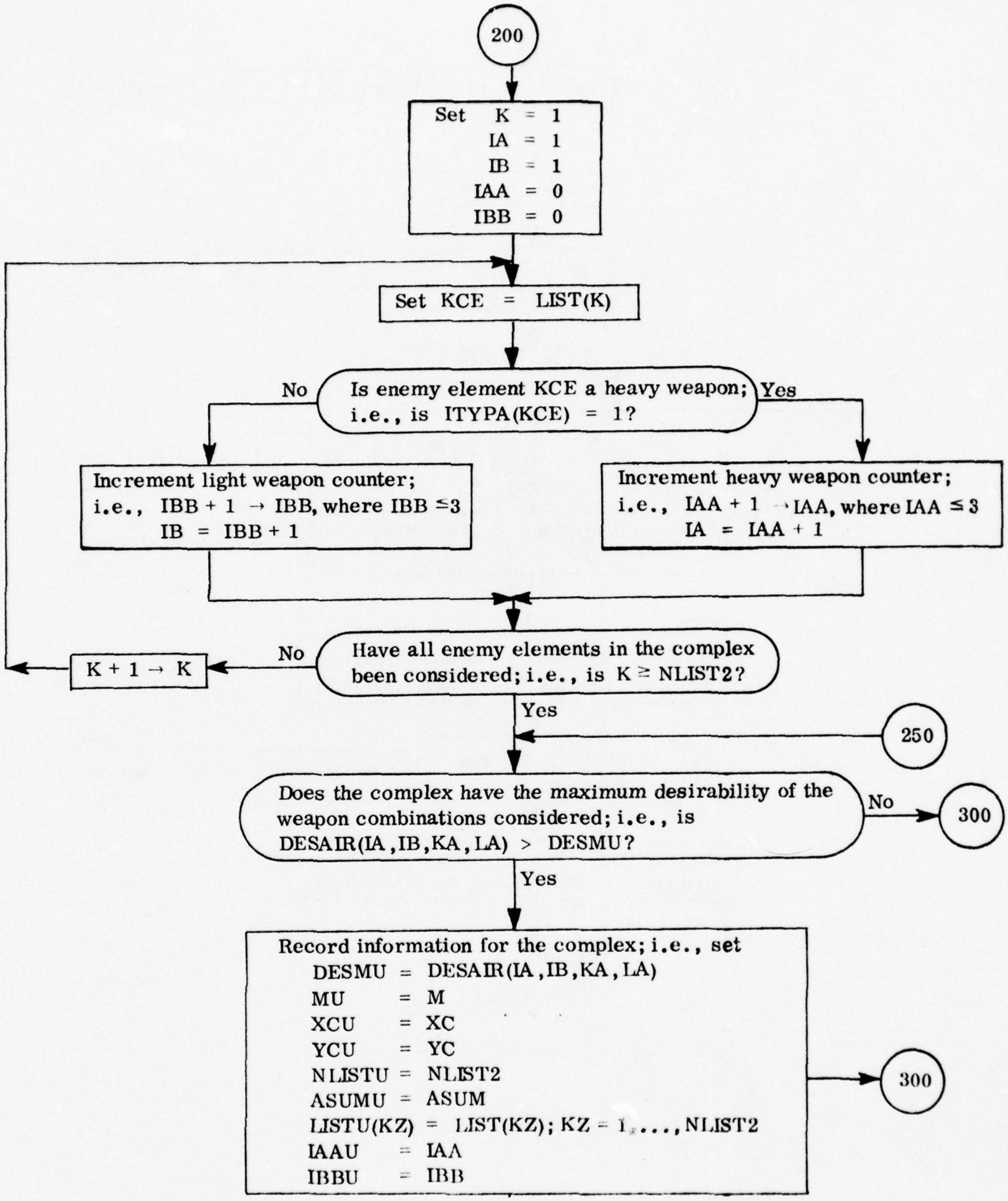
LENGTH: $10D8_{16} = 4312_{10}$ bytes



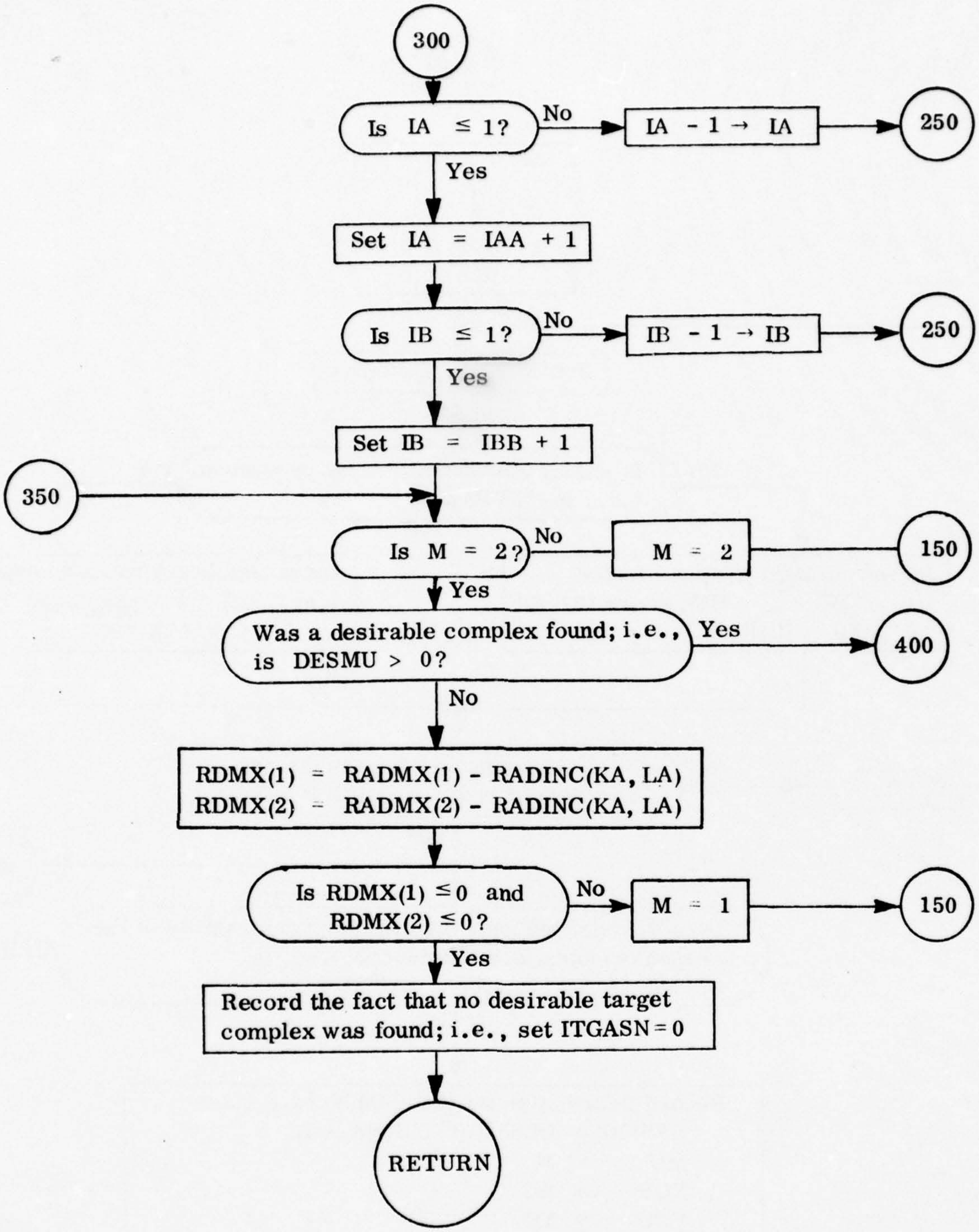
Subroutine AIRFIR: Direct-Fire Target Selection



Subroutine AIRFIR: Continued



Subroutine AIRFIR: Continued



Subroutine AIRFIR: Continued

400

Determine KELAGN(I, LHCE) and KAMAVL(I, LHCE), the enemy element number and ammunition code, respectively, assigned to aerial vehicle, LHCE, for gun system code, I, where

I = { 1 destructive weapon assignment; 2 first suppressive weapon assignment; 3 second suppressive weapon assignment }

by a call to subroutine WASAIR

Indicate a target assignment was made; i.e., ITGASN = 2

Initialize aerial firing model commons; i.e., call CONVRT

RETURN

Subroutine AIRFIR: Continued

Subroutine AIRPOR

PURPOSE: Subroutine AIRPOR determines the total priority weight for each enemy ground element known to a specified aerial vehicle section.

CALLING SEQUENCE:

CALL AIRPOR(IASC,IAE,IWC,NLIST,LIST,APRFCW, JJJ)

where

IASC	=	aerial vehicle section number
IAE	=	array of element numbers of section IASC
IWC	=	aerial vehicle section weapon code
NLIST	=	number of enemy elements known to elements of section IASC
LIST	=	array containing enemy element numbers known to elements of section IASC
APRFCW	=	array containing total priority weight for each enemy element in LIST.

JJJ = { 1 if priority weighting factors due to engagement by friendly air elements are to be ignored; ~1 otherwise

METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

ARFCW1	ARFCW7	ICECOM
ARFCW2	ARFCW8	ITYPA
ARFCW3	ARFCW9	LHICE
ARFCW4	BLUDET	LWCOD
ARFCW5	EFELC	MDFAF
ARFCW6	ETIM	NUMBER
		REDDET

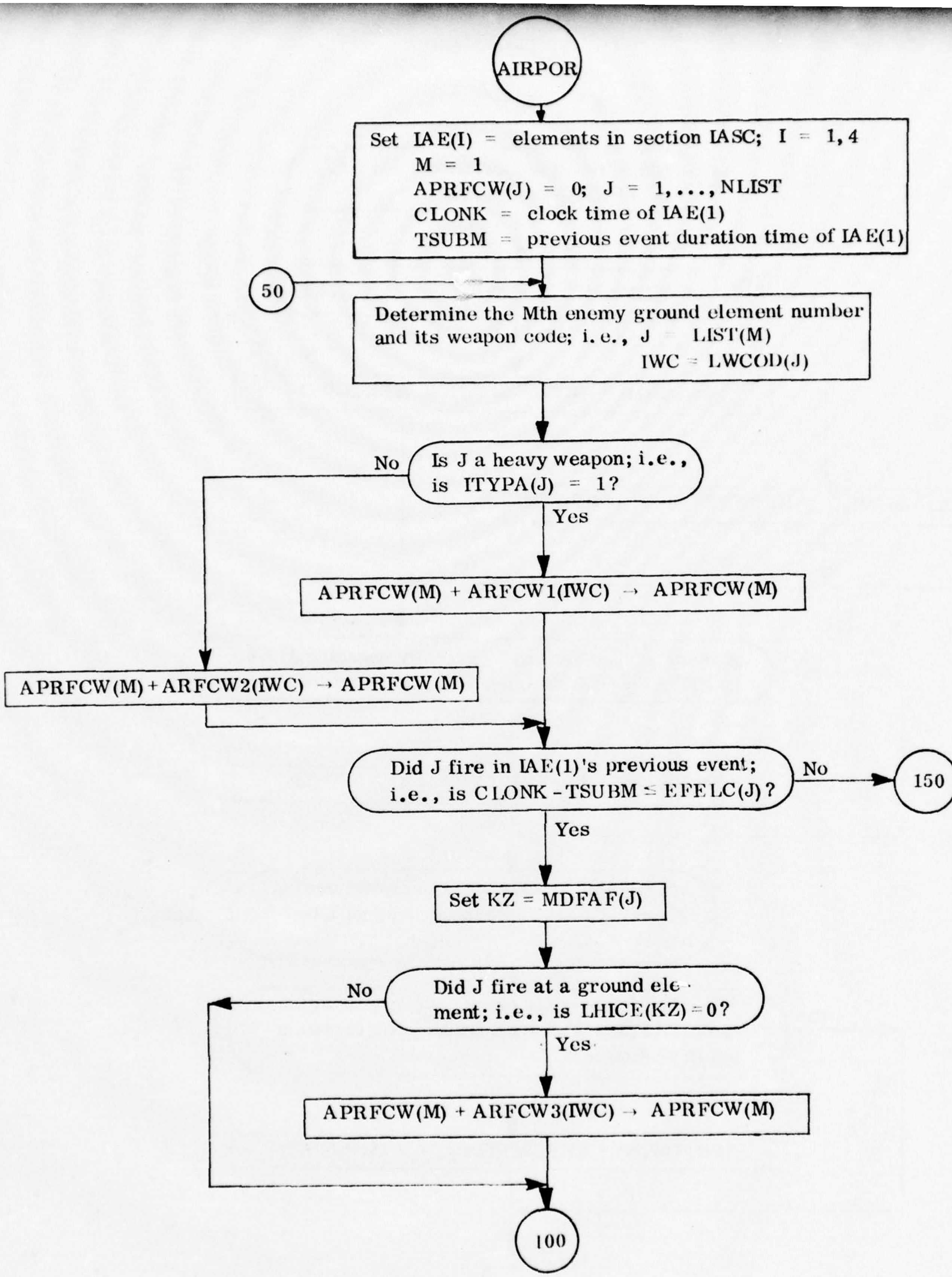
SUBROUTINES REQUIRED:

None

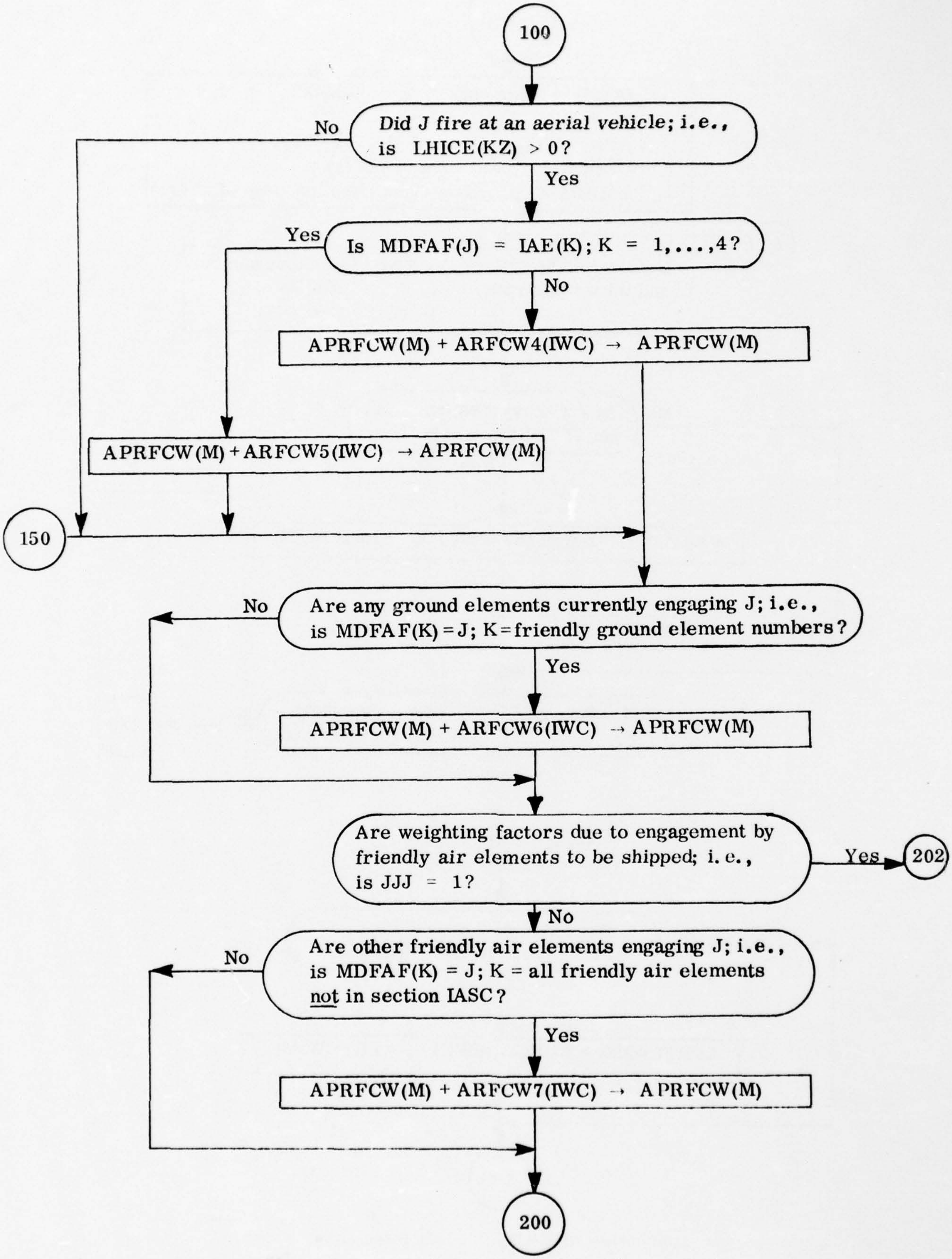
AIRPOR CALLED BY:

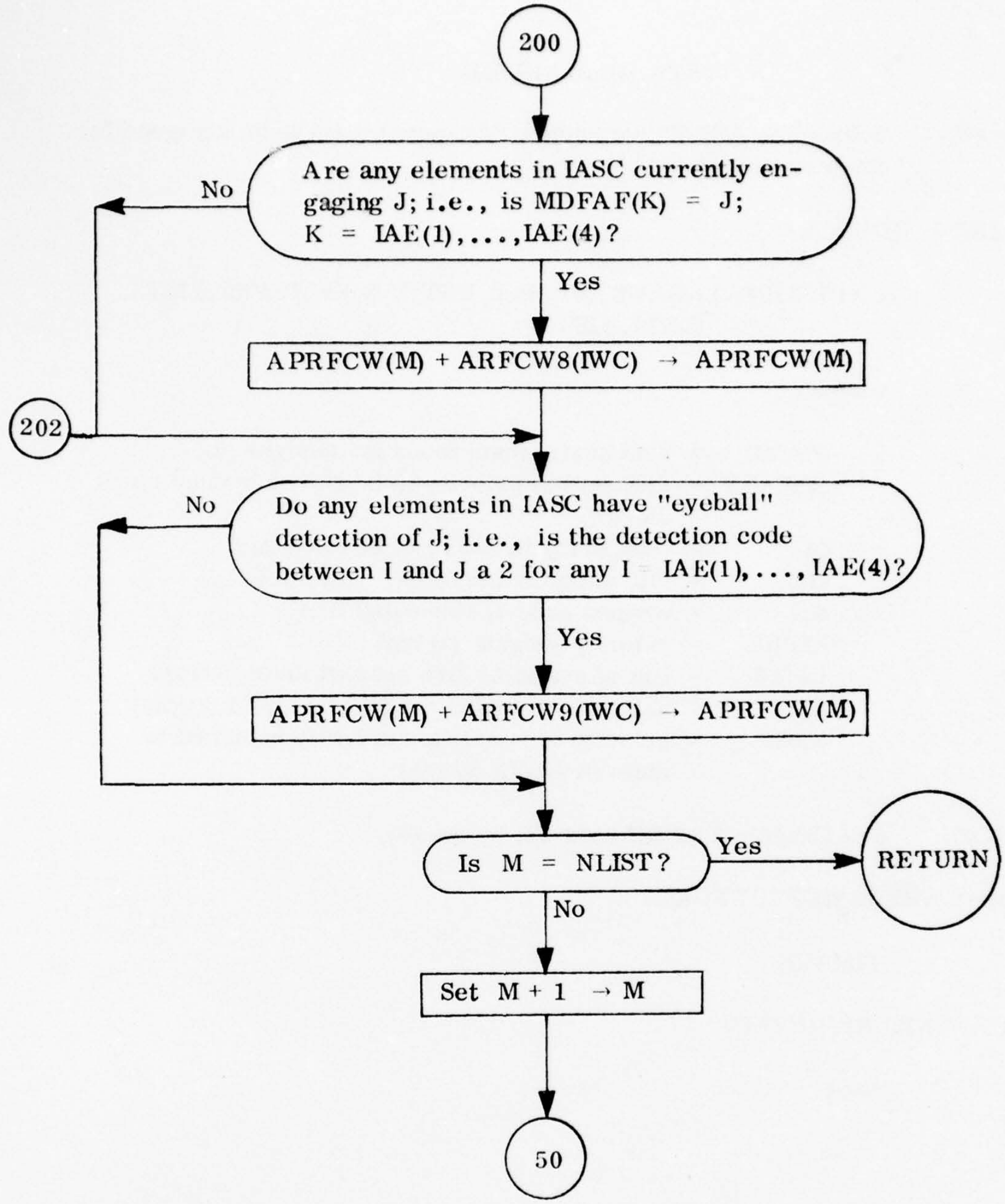
AIRFIR

LENGTH: $952_{16} = 2386_{10}$ bytes



Subroutine AIRPOR: Target Priorities


100
Did J fire at an aerial vehicle; i.e., is LHICE(KZ) > 0?
No
Yes
Is MDFAF(J) = IAE(K); K = 1,...,4?
Yes
No
APRFCW(M) + ARFCW4(IWC) → APRFCW(M)
APRFCW(M) + ARFCW5(IWC) → APRFCW(M)
150
Are any ground elements currently engaging J; i.e., is MDFAF(K) = J; K = friendly ground element numbers?
No
Yes
APRFCW(M) + ARFCW6(IWC) → APRFCW(M)
Are weighting factors due to engagement by friendly air elements to be shipped; i. e., is JJJ = 1?
Yes
202
No
Are other friendly air elements engaging J; i.e., is MDFAF(K) = J; K = all friendly air elements not in section IASC?
No
Yes
APRFCW(M) + ARFCW7(IWC) → APRFCW(M)
200



Subroutine AIRPOR: Continued

Subroutine AIRSEL

PURPOSE: Subroutine AIRSEL determines an aerial team to be assigned for mission execution.

CALLING SEQUENCE:

```
CALL AIRSEL(JSAVE,IPRMAX,INIT,IL2,KK,KIPRS,LIST3,
            LIST4,LIST5)
```

where

JSAVE	=	best aerial team found for assignment
IPRMAX	=	differential priority to be gained by assigning JSAVE
INIT	=	first entry in LIST3 to be examined
IL2	=	last entry in LIST3 to be examined
KK	=	weapon code of requested fire
KIPRS	=	priority weights (array)
LIST3	=	list of available fire support units (array)
LIST4	=	unit weapon codes of units in LIST3 (array)
LIST5	=	priority of missions currently assigned to units in LIST3 (array)

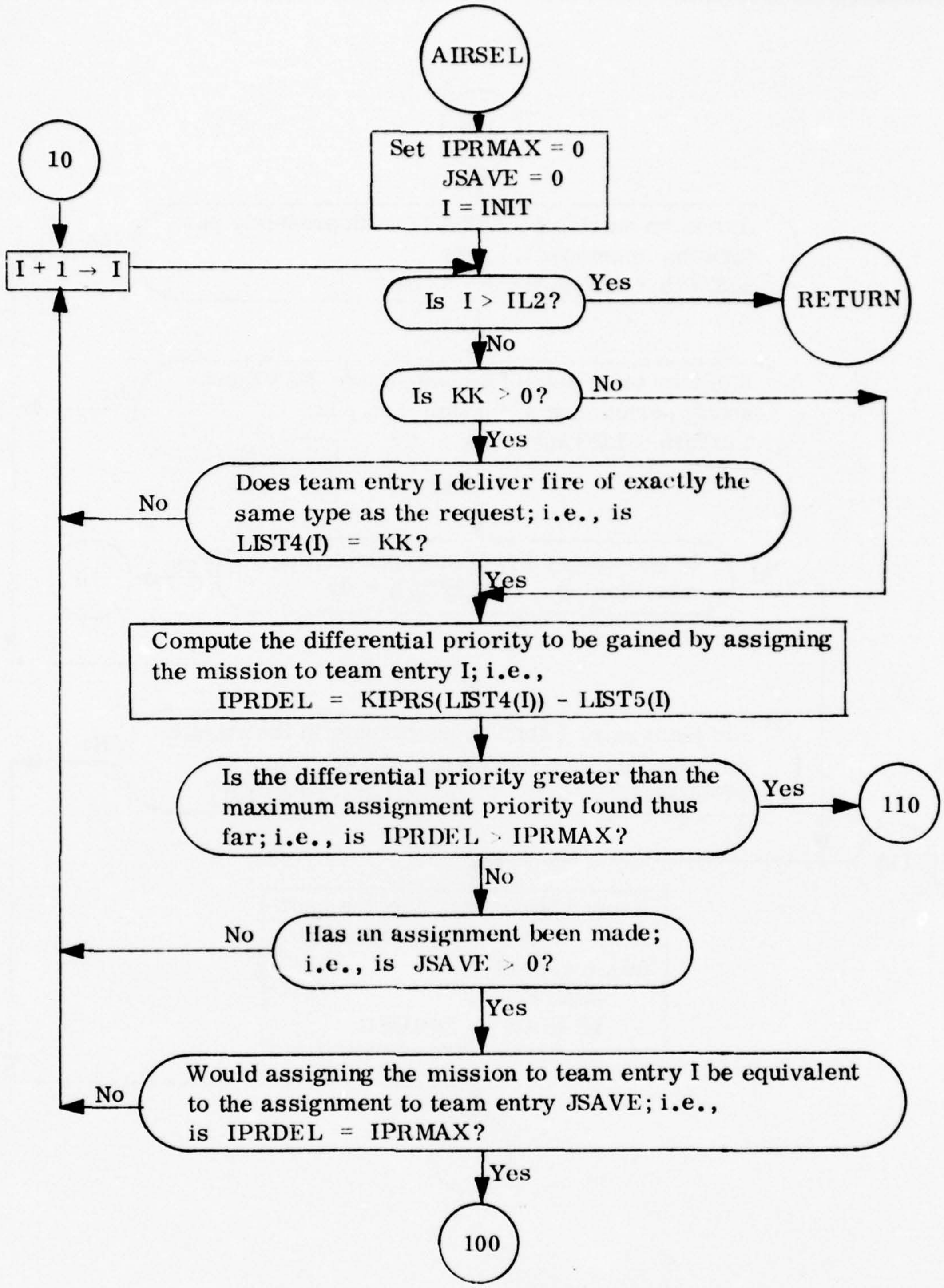
METHOD: See Chapter 2 of RF 2978 FR 71-3A (U).

COMMON AREAS REFERENCED:

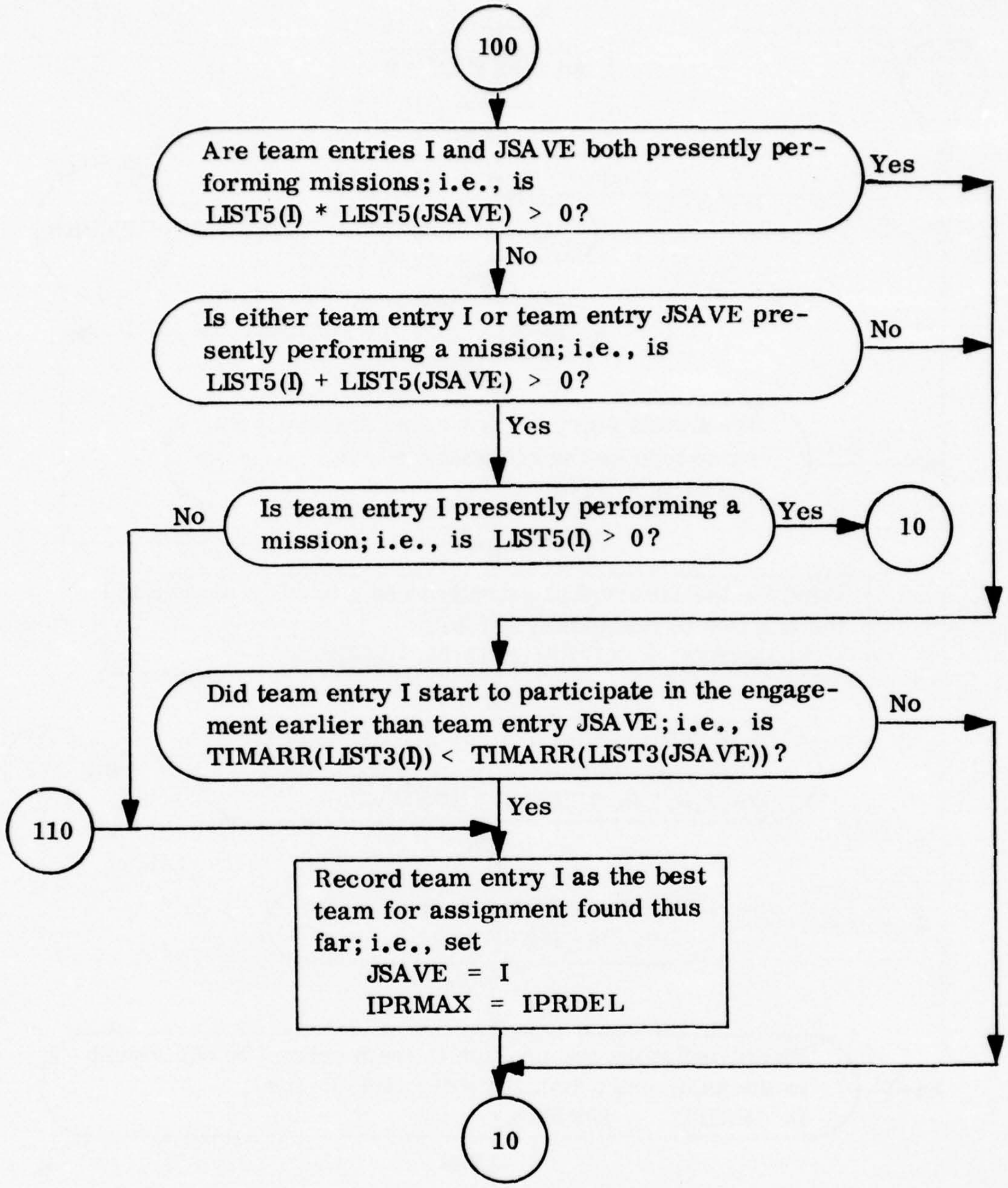
TIMARR

SUBROUTINES REQUIRED:

None



Subroutine AIRSEL: Aerial Unit Selection



Subroutine AIRSEL: Continued

Subroutine APPINT

PURPOSE: Subroutine APPINT is used to store and to retrieve aerial vehicle movement data. A movement data set consists of position, velocity and acceleration components in the battlefield X, Y and Z directions.

CALLING SEQUENCE:

CALL APPINT(FLAG, TIME, NSECL)

where

FLAG
- = 0 indicates movement information is to be retrieved for an aerial vehicle
- > 0 indicates movement information is to be stored for an aerial vehicle where the number of the data set to be stored is specified by FLAG

TIME =
- current aerial vehicle's clock time if FLAG > 0
- clock time at which information is desired if FLAG = 0

NSECL = number of the aerial vehicle's section.

METHOD: During helicopter movement events, position, speed and acceleration components in the X, Y and Z directions are stored for the current element at one second intervals. The storage arrays are arranged so that the present movement data set always appears first in the arrays. Storage is accomplished by calling subroutine APPINT from the movement subroutine APFDYS. When position information is desired for a helicopter that is not the current element (e.g., during air defense weapon firing events), subroutine APPINT is used to determine interpolated data from the storage arrays. Acceleration components in the X, Y and Z directions are determined by linear interpolation while speed and position components are determined by the standard relations

$V = V_0 + At$ and $S = S_0 + Vt + 1/2At^2$,

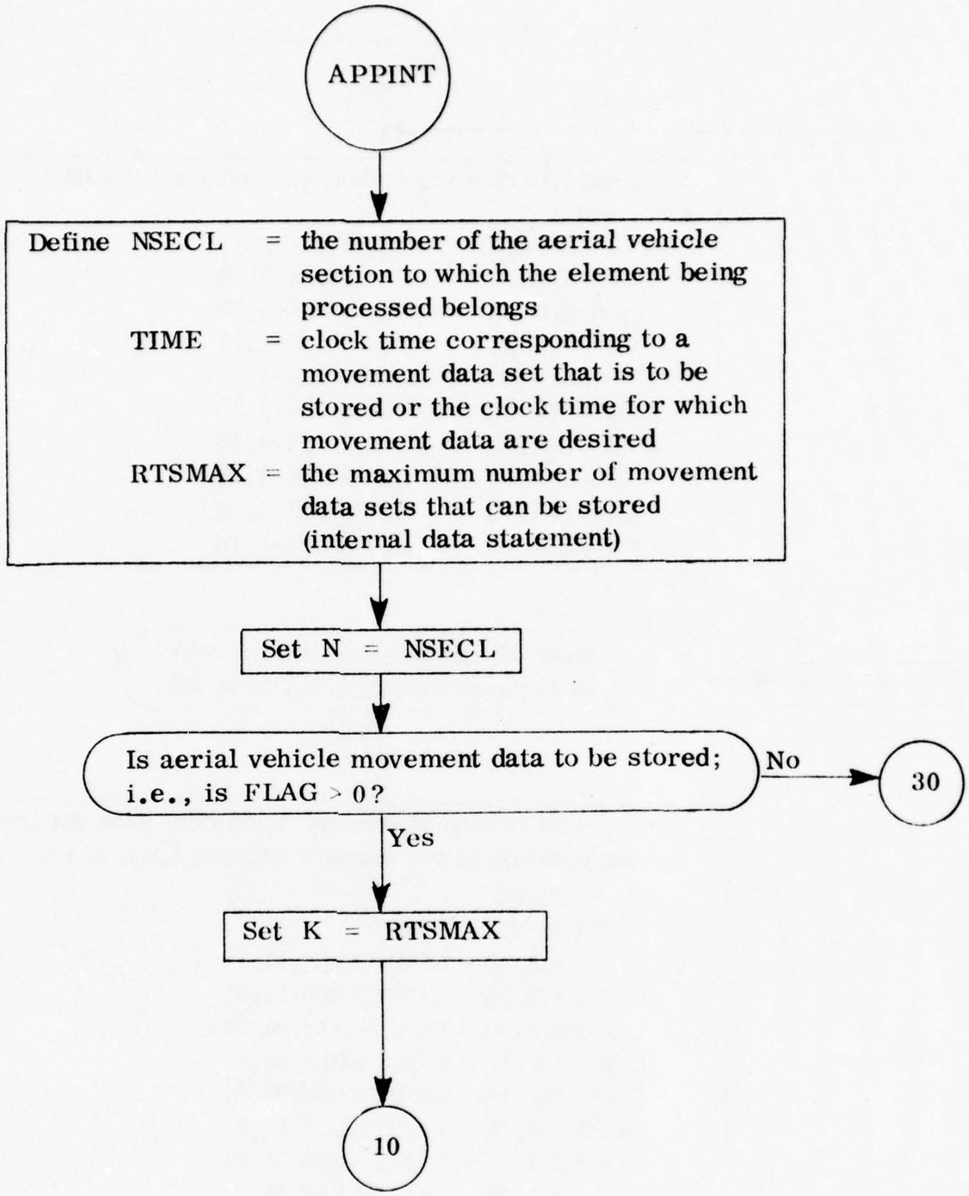
respectively. When the storage arrays become full, the oldest values are deleted and the newest values are added.

COMMON AREAS REFERENCED:

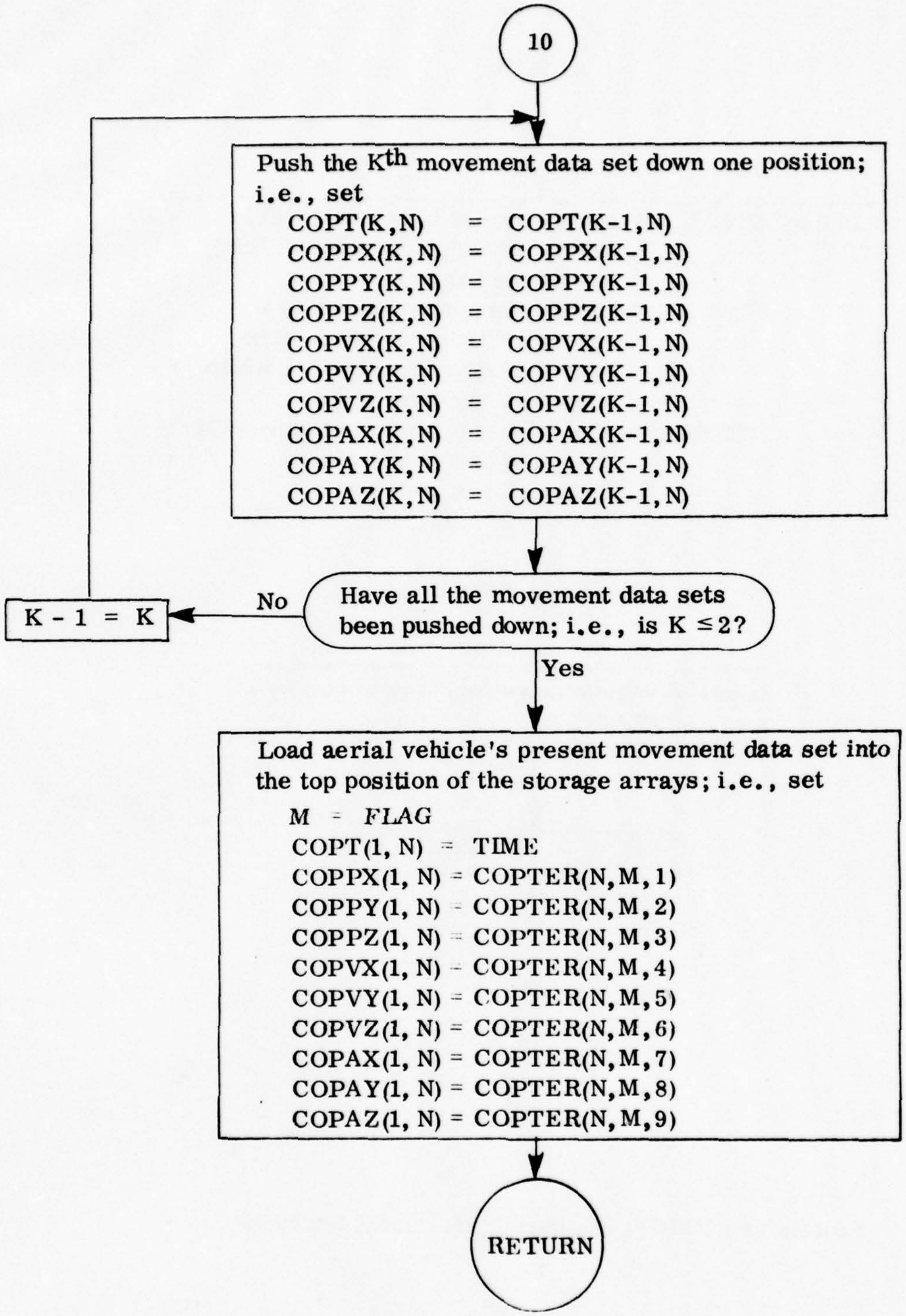
COPAX	COPTER	COPVX
COPAY	COPPX	COPVY
COPAZ	COPPY	COPVZ
COPT	COPPZ	

SUBROUTINES REQUIRED:

None



Subroutine APPINT: Helicopter Movement Interpolation



Subroutine APPINT: Continued

30

Initialize for search of time array to locate interpolation interval; i.e., set K = 1
Set I = MNSEC + 1

Is COPT(K,N) ≥ TIME and COPT(K + 1,N) ≤ TIME?

Yes

No

Has the entire array been searched; i.e., is K = RTSMAX

No

K = K + 1

Yes

60

Determine ΔT, the time increment within the interpolation interval; i.e., set

DT = TIME - COPT(K + 1, N)

Set the acceleration values in the X, Y, and Z directions; i.e., set

COPTER(I, 7, 7) = [COPAX(K, N)-COPAX(K+1, N)]*TI+COPAX(K+1, N)
COPTER(I, 7, 8) = [COPAY(K, N)-COPAY(K+1, N)]*TI+COPAY(K+1, N)
COPTER(I, 7, 9) = [COPAZ(K, N)-COPAZ(K+1, N)]*TI+COPAZ(K+1, N)

where

$$TI = \frac{DT}{(COPT(K,N) - COPT(K + 1,N))}$$

70

Subroutine APPINT: Continued

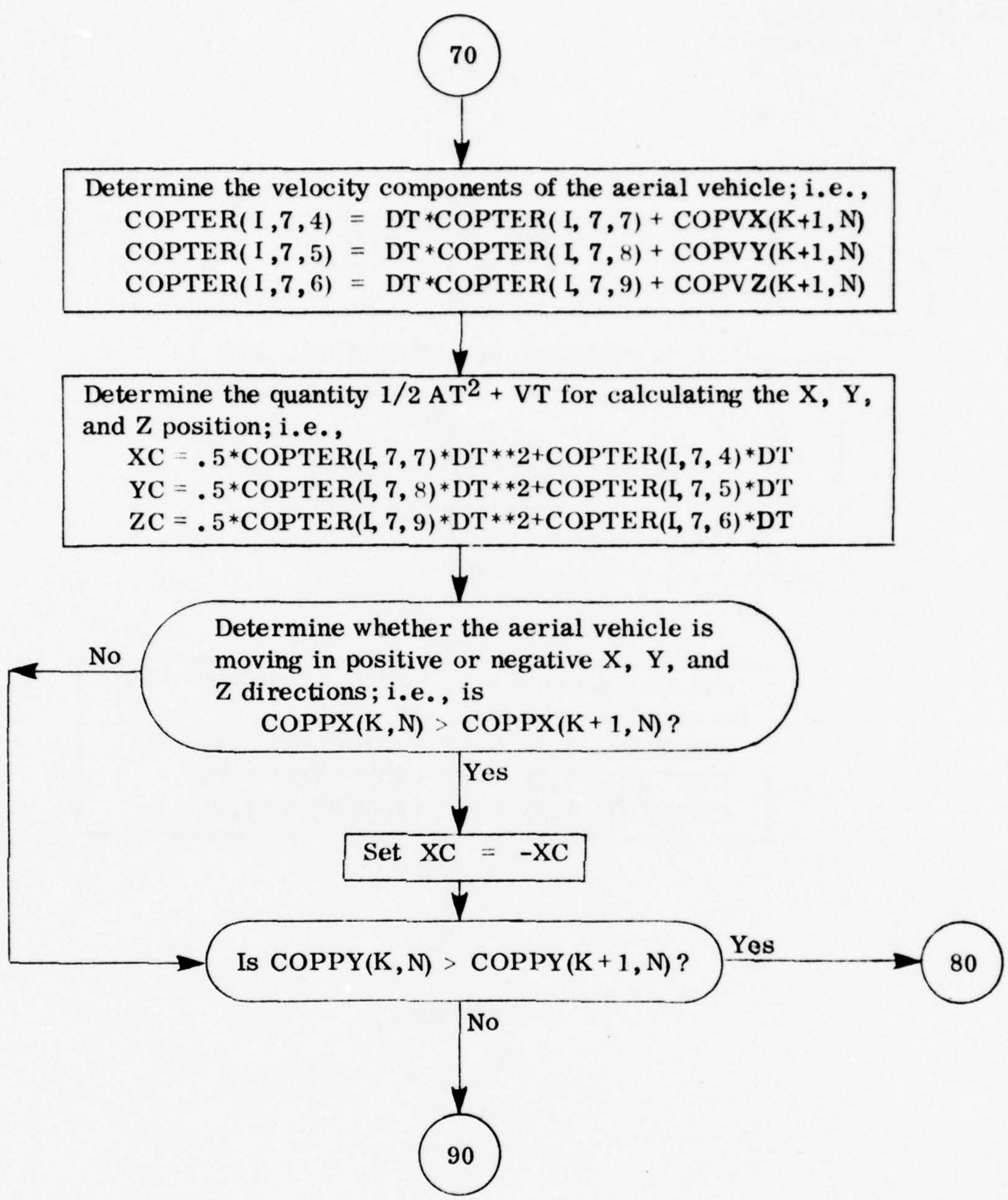
60

Load all values to correspond to the earliest values saved; i.e., set

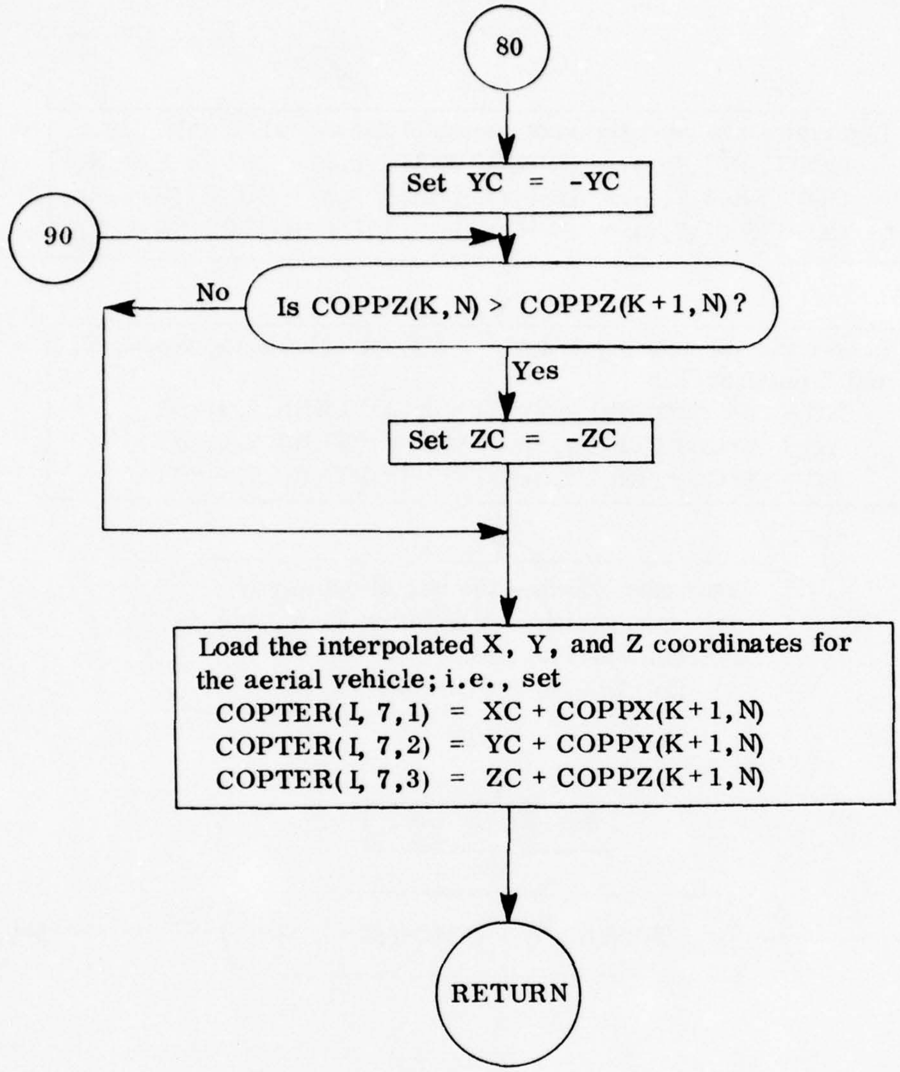
COPTER(I, 7,1) = COPPX(K, N)
COPTER(I, 7,4) = COPVX(K, N)
COPTER(I, 7,7) = COPAX(K, N)
COPTER(I, 7,2) = COPPY(K, N)
COPTER(I, 7,5) = COPVY(K, N)
COPTER(I, 7,8) = COPAY(K, N)
COPTER(I, 7,3) = COPPZ(K, N)
COPTER(I, 7,6) = COPVZ(K, N)
COPTER(I, 7,9) = COPAZ(K, N)

RETURN

Subroutine APPINT: Continued



Subroutine APPINT: Continued



Subroutine APPINT: Continued

Subroutine ARFO

PURPOSE: Subroutine ARFO is used to control the indirect-fire activities of launchers and forward observers in the presence of direct-fire activities.

CALLING SEQUENCE:

CALL ARFO

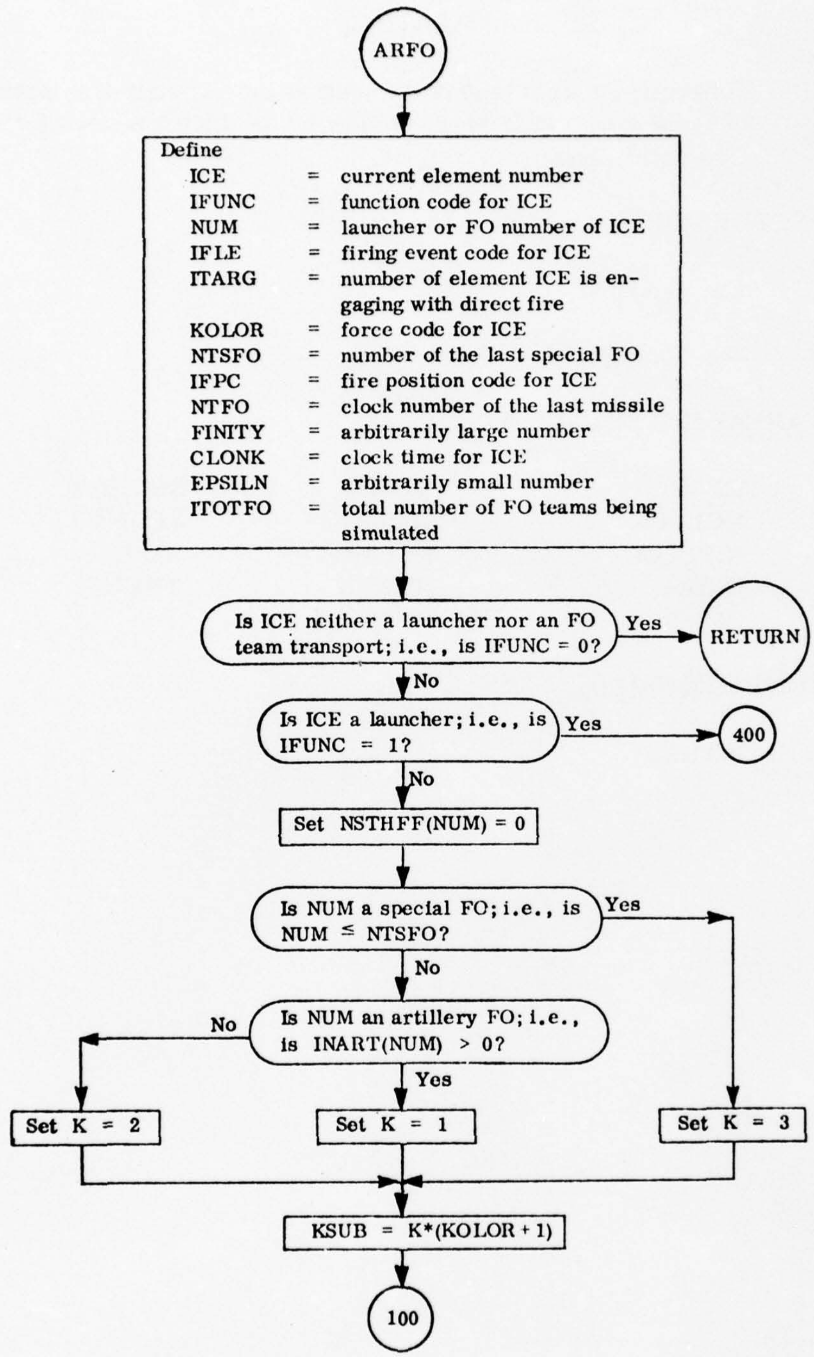
METHOD: See Chapter 9 of Volume 1.

COMMON AREAS REFERENCED:

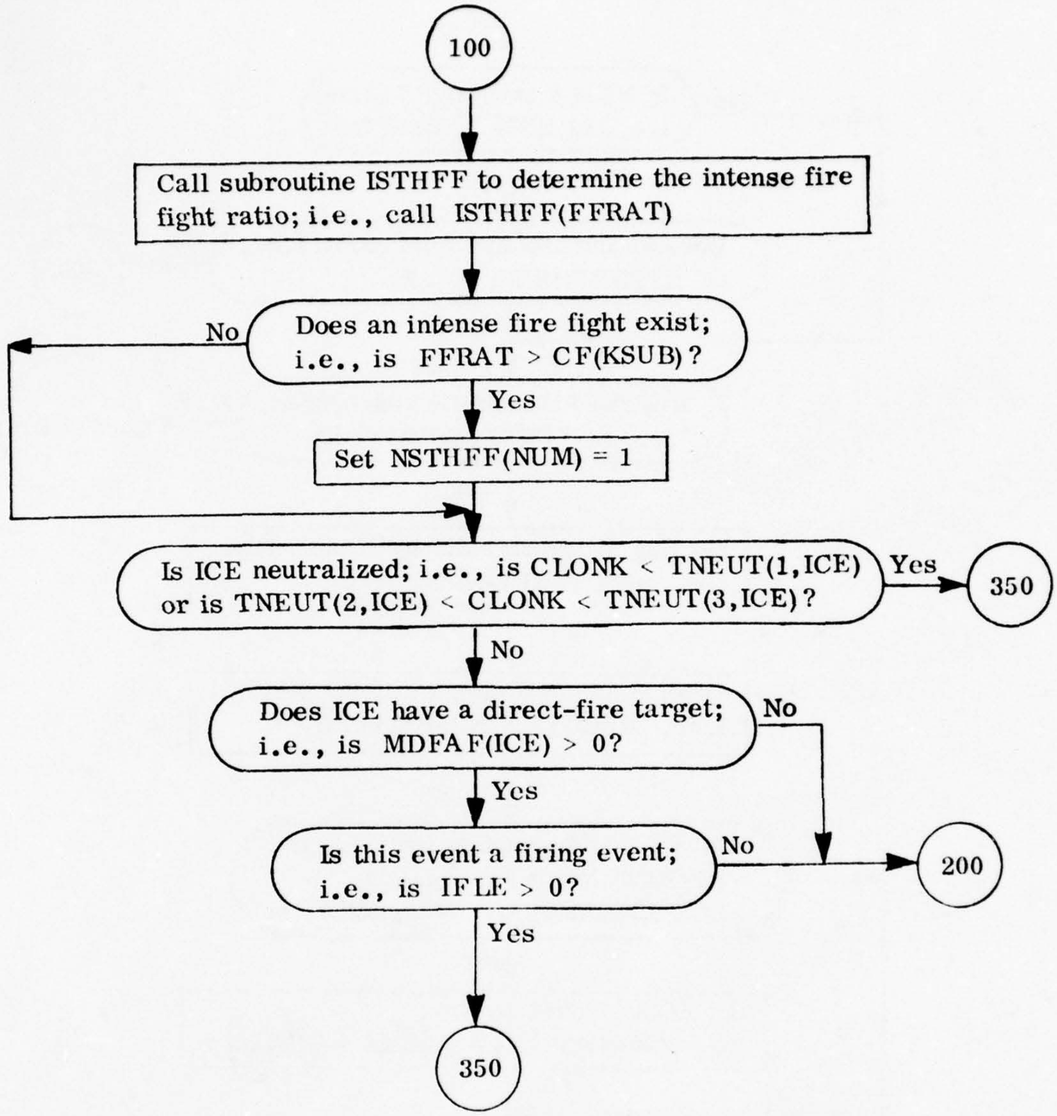
CF	KFOD	SEQPAR
ECLOCK	MDFAF	SPDMU
ICECOM	NSTHFF	TINIT
IFRFL	NTELE	TNEUT
INART	NUMBER	

SUBROUTINES REQUIRED:

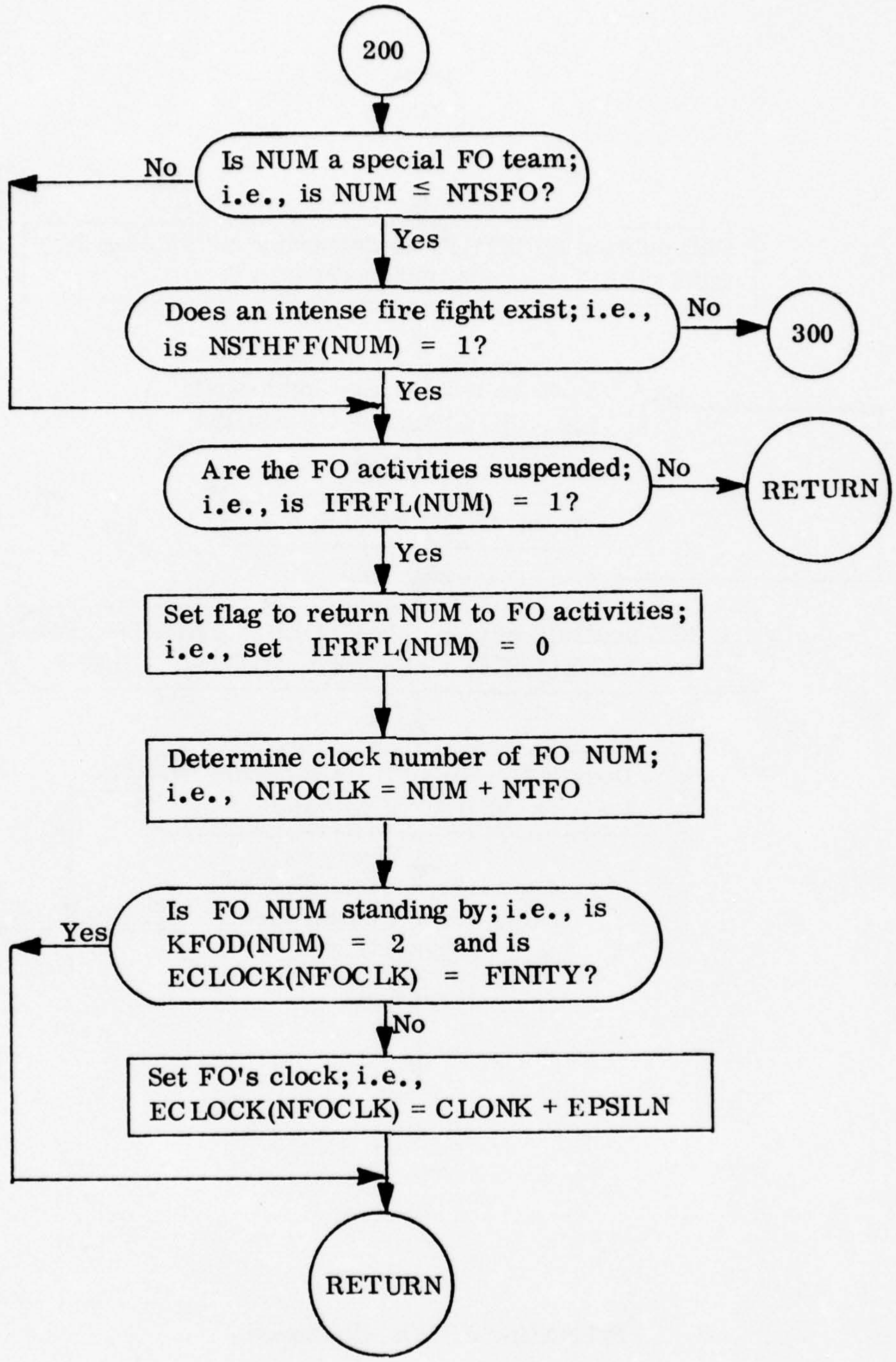
ISTHFF



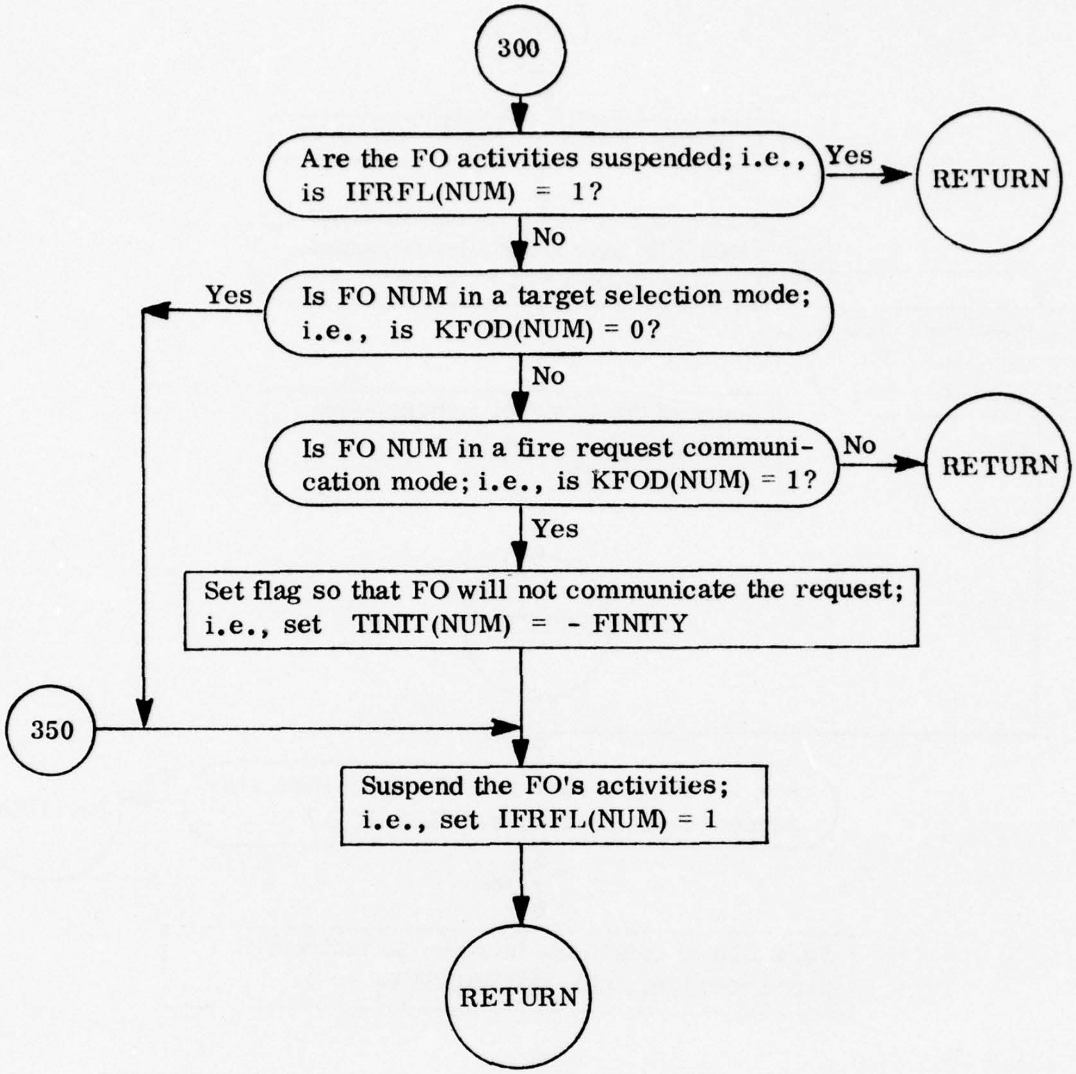
Subroutine ARFO: Launcher and Forward Observer Activity Control



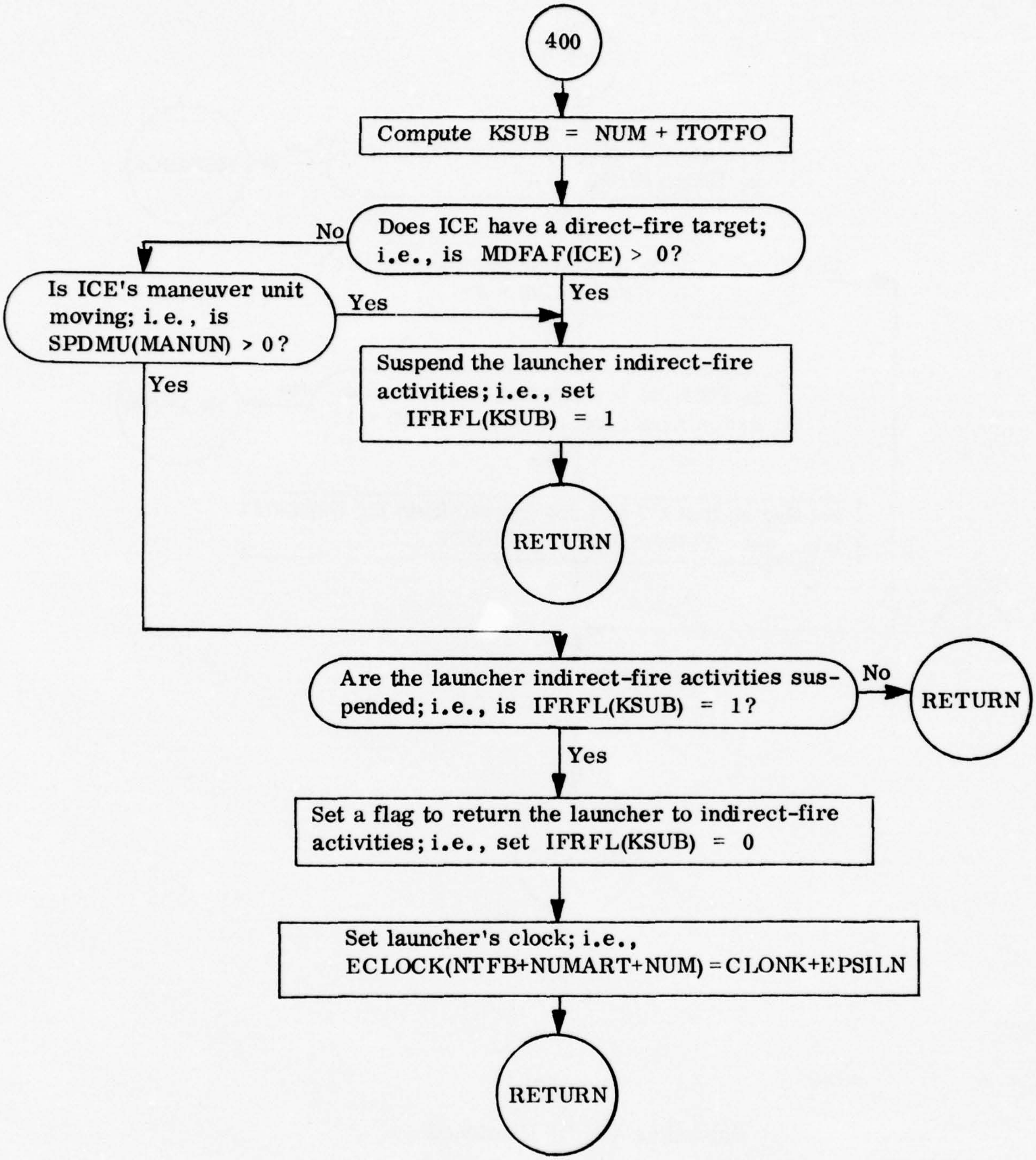
Subroutine ARFO: Continued



Subroutine ARFO: Continued



Subroutine ARFO: Continued



Subroutine ARFO: Continued

Subroutine ATAN5

PURPOSE: On numerous occasions with subroutine INTELL, it is desired to calculate the angle THETAC from an observer to a target (measured in a horizontal plane and in a counterclockwise sense) and to compare deviations of THETAC from DIR and from THETA0 with permissible deviations, ANGLIM and CKANG, respectively. Subroutine ATAN5 is designed to determine THETAC and to selectively perform the specified logical comparisons.

CALLING SEQUENCE:

CALL ATAN5 (X,Y,DIR,THETAC,L1,L2,KEY)

where

(X,Y) = battlefield coordinates of the target element

DIR = direction of travel of the observing element

THETAC = the angle that a line from the observing element to the target element makes with the battlefield X axis (measured in a horizontal plane and in a counterclockwise sense)

L1 = {
- logical true/false value returned to indicate results of logical comparisons
- TRUE if comparison is made and it is found that deviation of THETAC from DIR is greater than ANGLIM
- FALSE otherwise

L2 = {
- logical true/false value returned to indicate results of logical comparisons
- TRUE if comparison is made and it is found that deviation of THETAC from THETA0 is greater than CKANG
- FALSE otherwise

KEY = { input parameter to control logical operations of ATAN5
- 0 if all logical operations are to be executed
- 1 if angular deviation is not to be compared with CKANG
- 2 if angular deviation is not to be compared with ANGLIM
- 3 if no comparisons are to be made

RESTRICTIONS: See CALLING SEQUENCE.

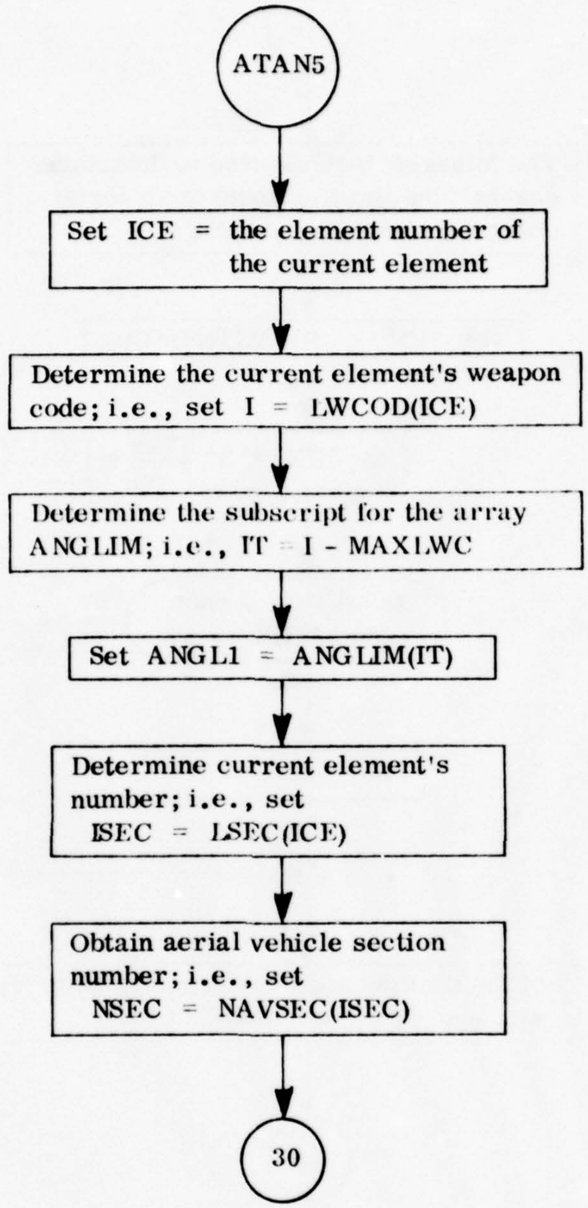
METHOD: After the angle of the line from the observing element to the target element has been calculated, the parameter KEY is examined to determine what logical operations are to be performed. If the parameter KEY indicates comparisons are to be made, a nonstandard return is made whenever it is found that either of the angular deviations exceed the permissible values.

COMMON AREAS REFERENCED:

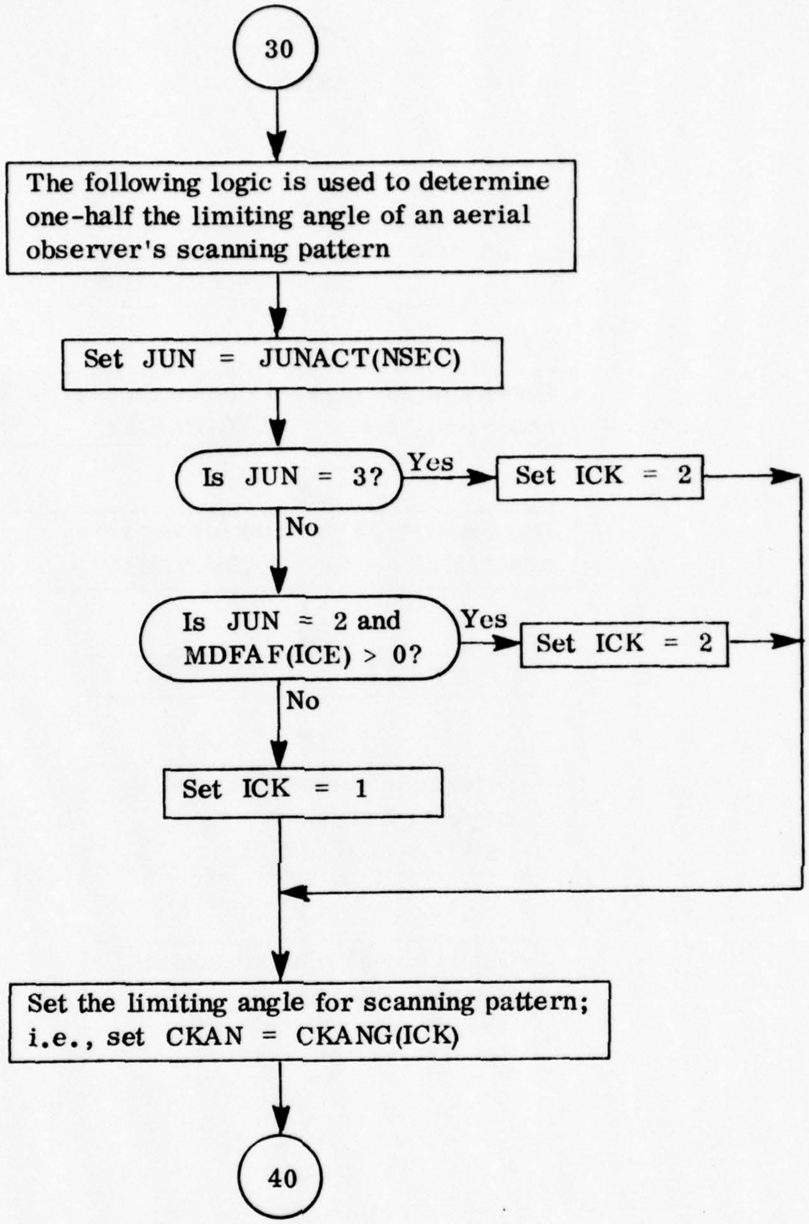
ANGLIM	JUNACT	NAVSEC
CKANG	LWCOD	NUMBER
ICECOM	MDFAF	THETA0
JPHASE		

SUBROUTINES REQUIRED:

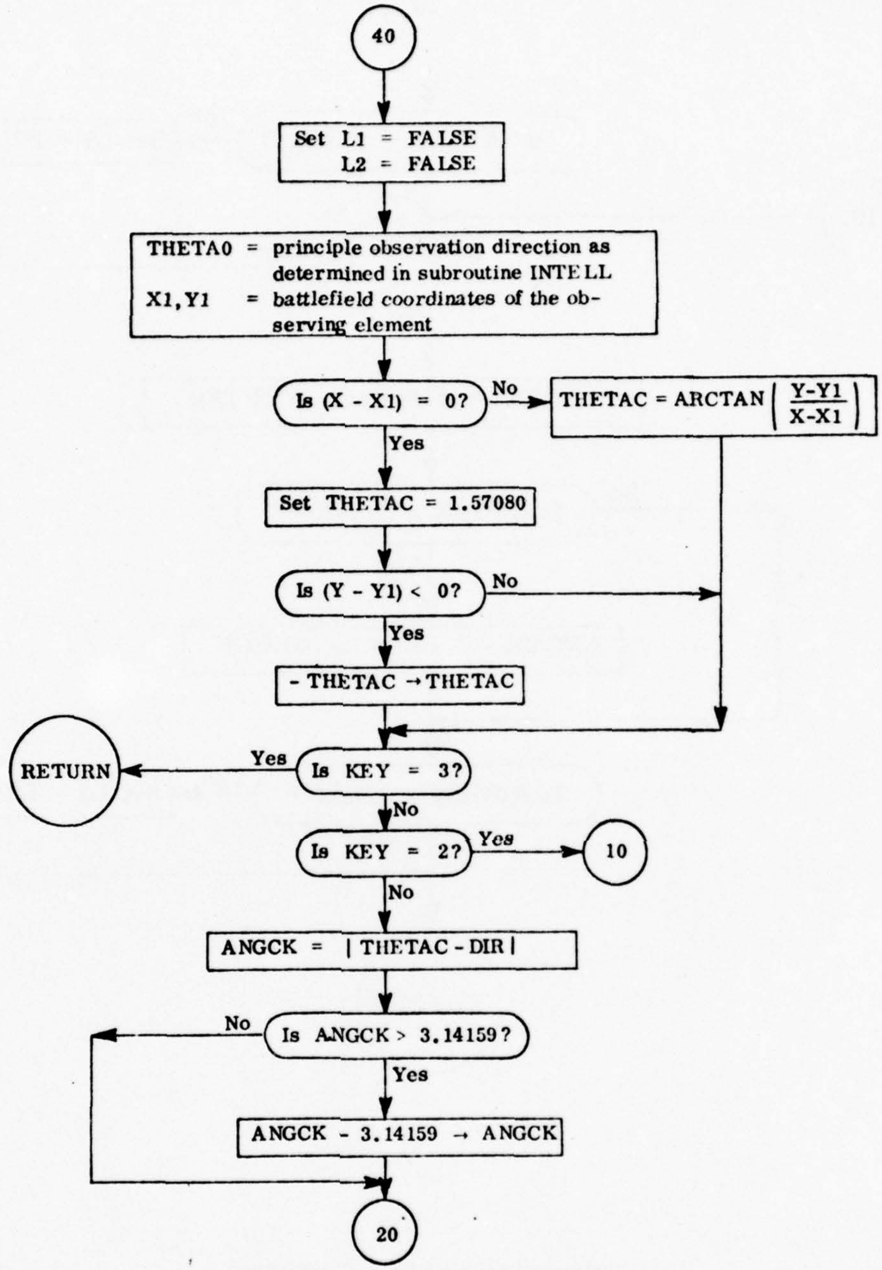
ATAN2



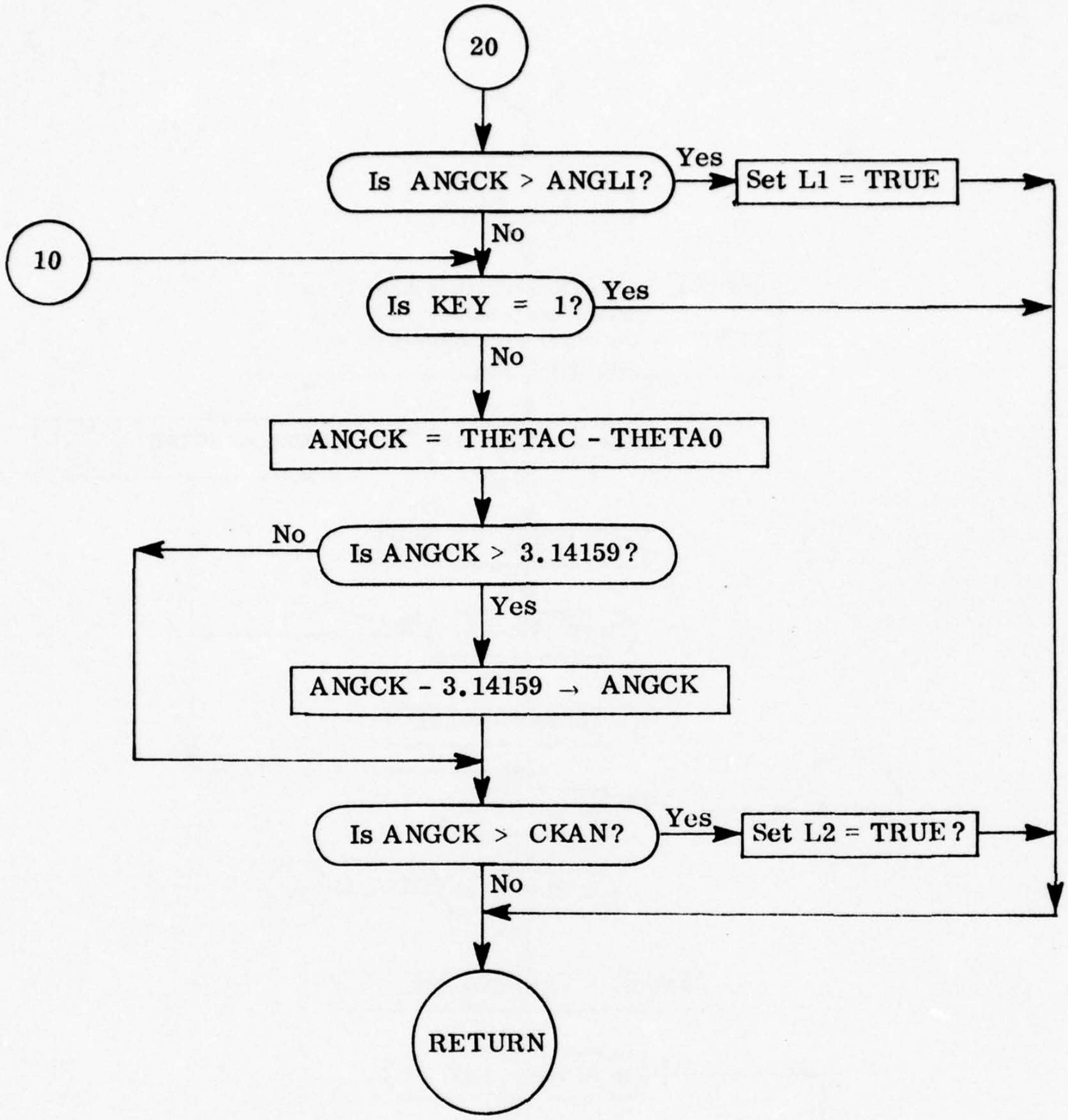
Subroutine ATAN5: Angle Comparisons for Subroutine INTELL



Subroutine ATAN5: Continued



Subroutine ATAN5: Continued



Subroutine ATAN5: Continued

Subroutine ATDEC

PURPOSE: Subroutine ATDEC is used to make movement decisions for an aerial maneuver unit.

CALLING SEQUENCE:

CALL ATDEC(NAT,KDEC)

where

NAT = aerial maneuver unit number for which a movement decision is desired

KDEC = parameter indicating movement decision made
- 0 no decision has been made
- 1 unit should retire
- 2 unit should seek a defensive position
- 3 unit should commence a counter-measure mission
- 4 unit should terminate its present mission and stand by for reassignment
- 5 unit should move its indirect-fire MISTIC loiter station to a new position

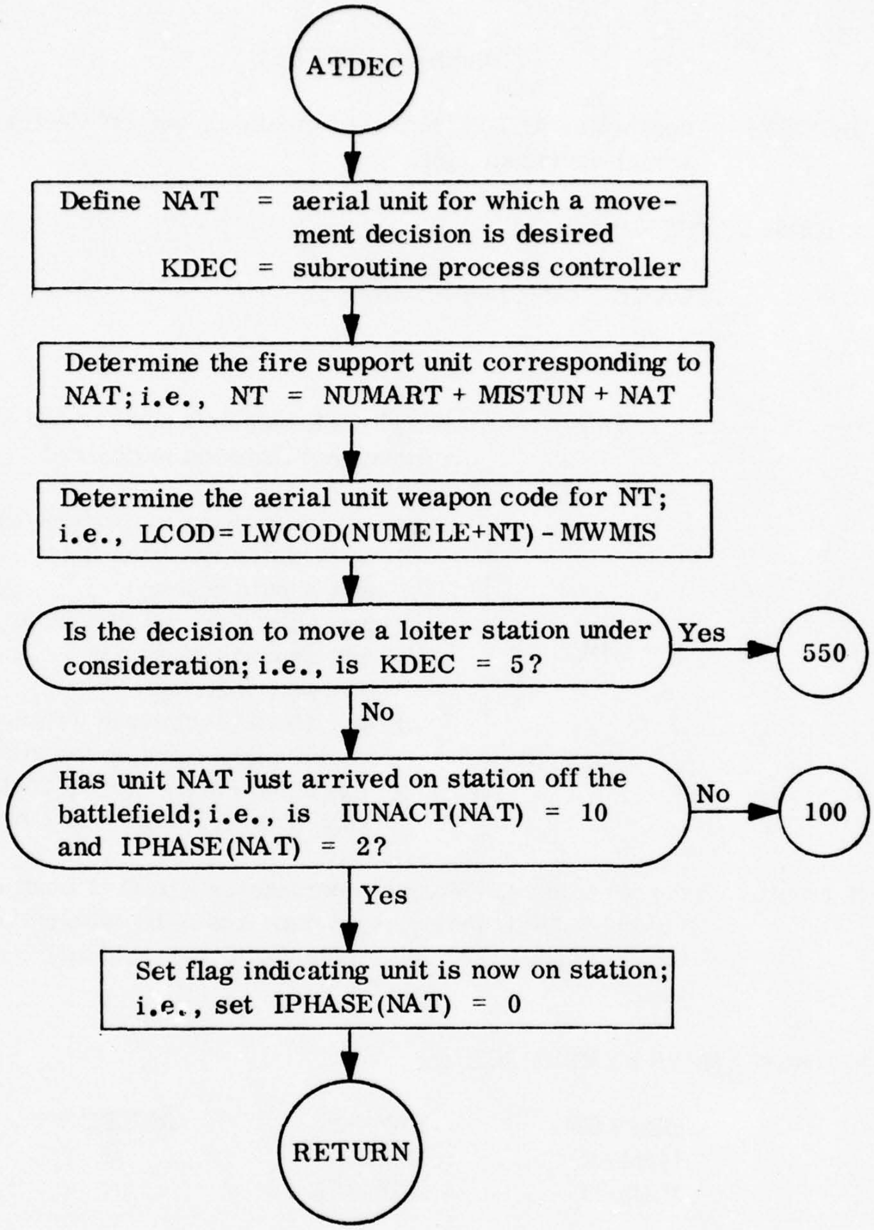
METHOD: See Chapter 4. When the parameter KDEC is input as 0, the value of KDEC may remain 0 or it may be changed to 1, 2, 3 or 4. When KDEC is input as 5, it may remain 5 or be changed to 0.

COMMON AREAS REFERENCED:

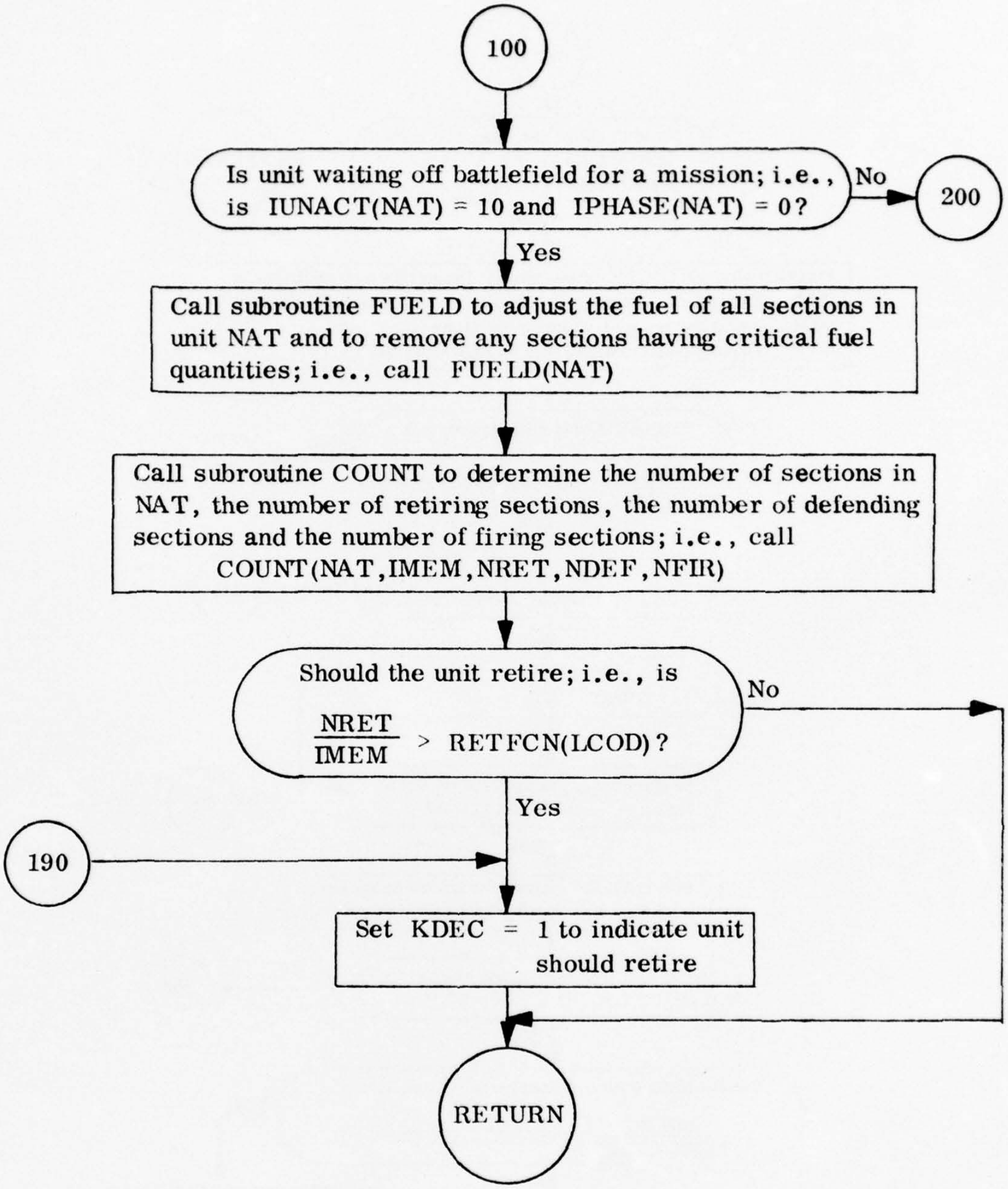
DEFFCN	LWCOD	RETFCN
IPHASE	MCLASS	
IUNACT	NUMBER	

SUBROUTINES REQUIRED:

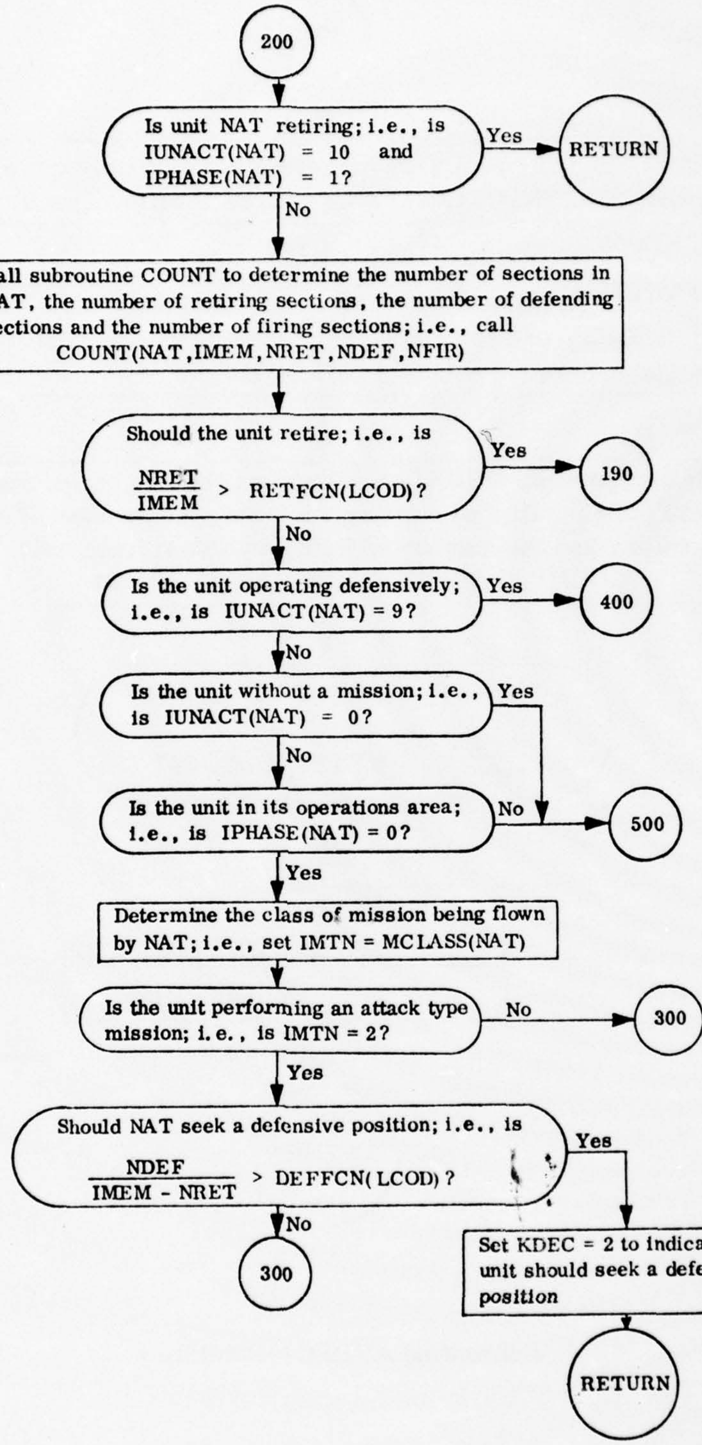
FUELD	CMMIS	LOIMOV
COUNT	MISEND	



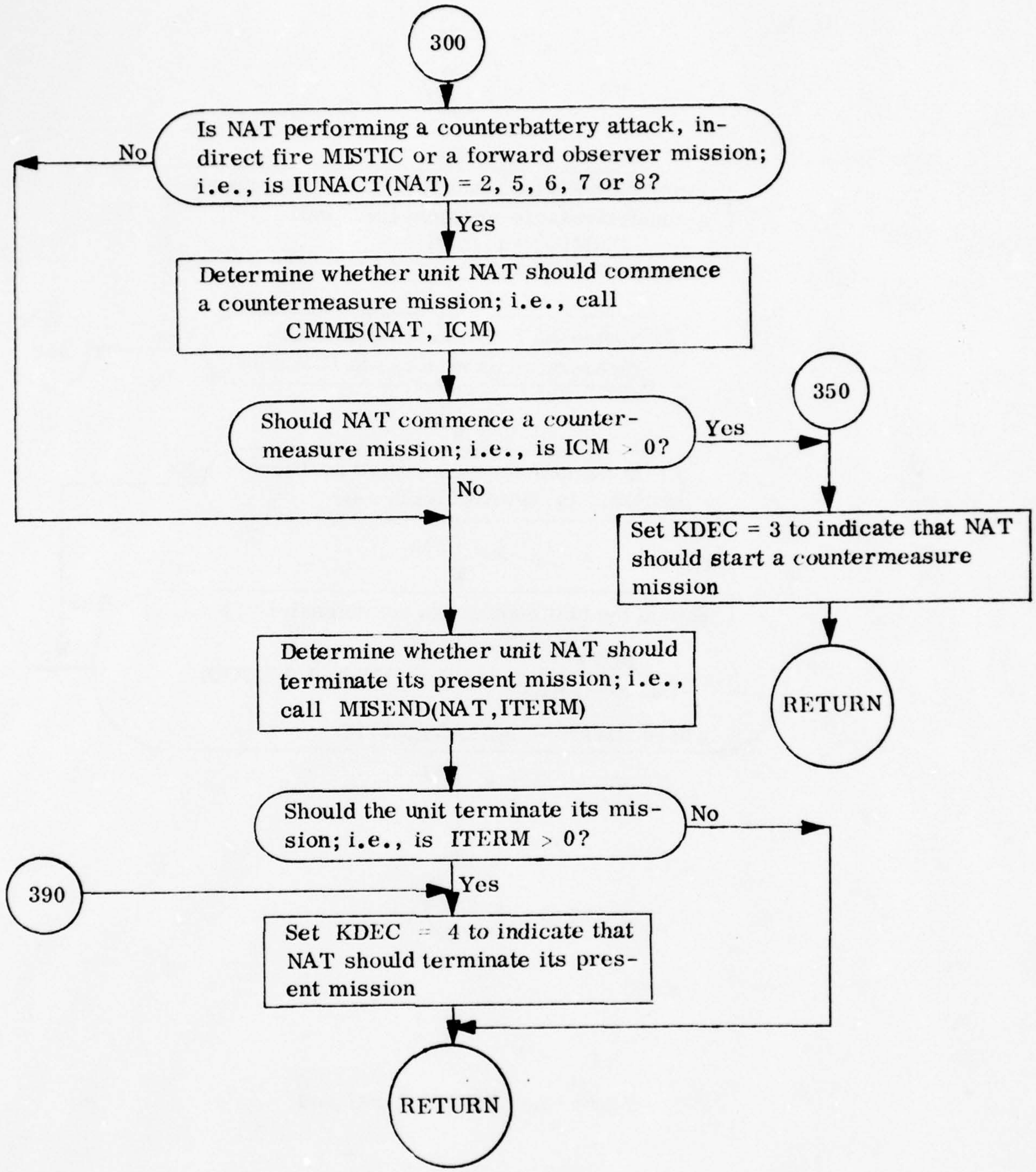
Subroutine ATDEC: Aerial Unit Mission Termination Decisions



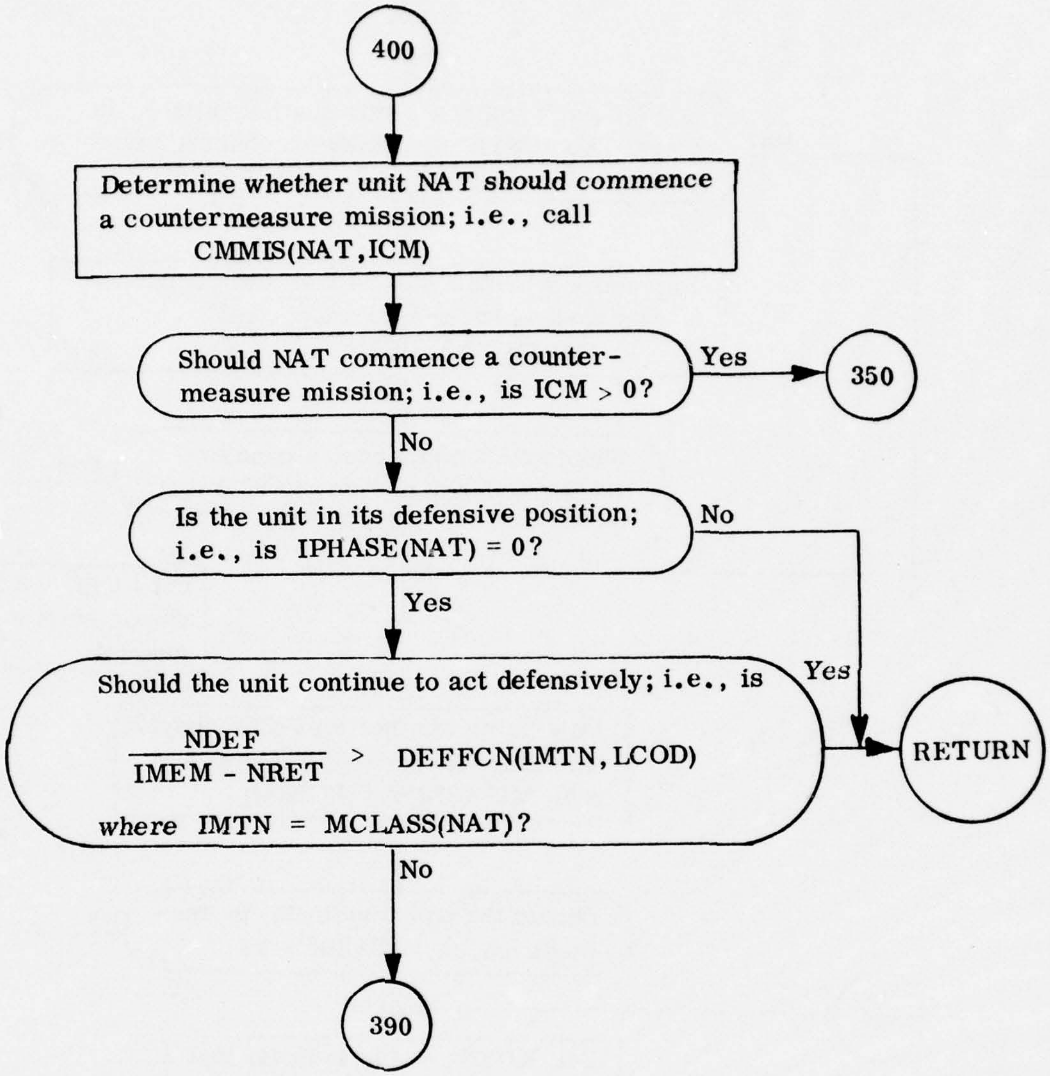
Subroutine ATDEC: Continued



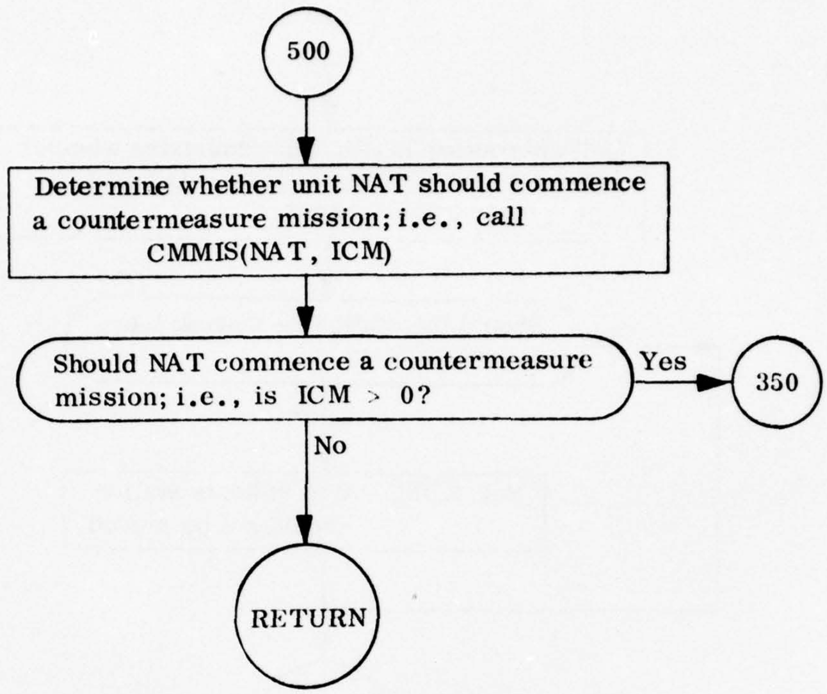
Subroutine ATDEC: Continued



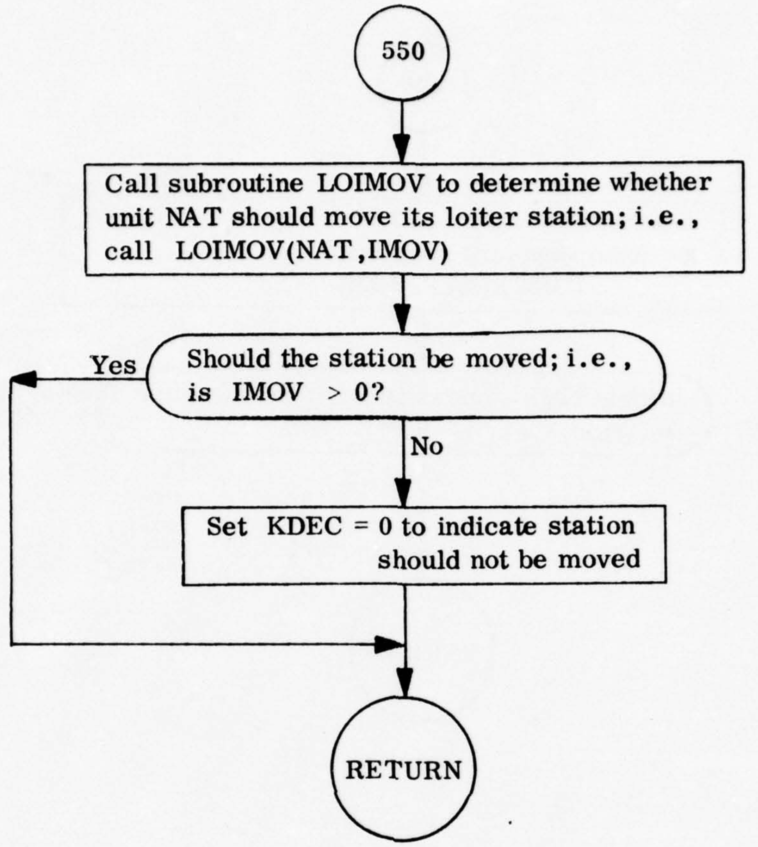
Subroutine ATDEC: Continued



Subroutine ATDEC: Continued



Subroutine ATDEC: Continued



Subroutine ATDEC: Continued

Subroutine ATKPRM

PURPOSE: Subroutine ATKPRM records data to be used by the aerial vehicle firing model.

CALLING SEQUENCE:

CALL ATKPRM(ICNT,IRS)

where

ICNT = { 1 if the first phase of an aerial attack is being analyzed; 2 if the second phase of an aerial attack is being analyzed }

IRS = indicator for the route to be selected as a result of the attack (see subroutine PICKRT)

METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

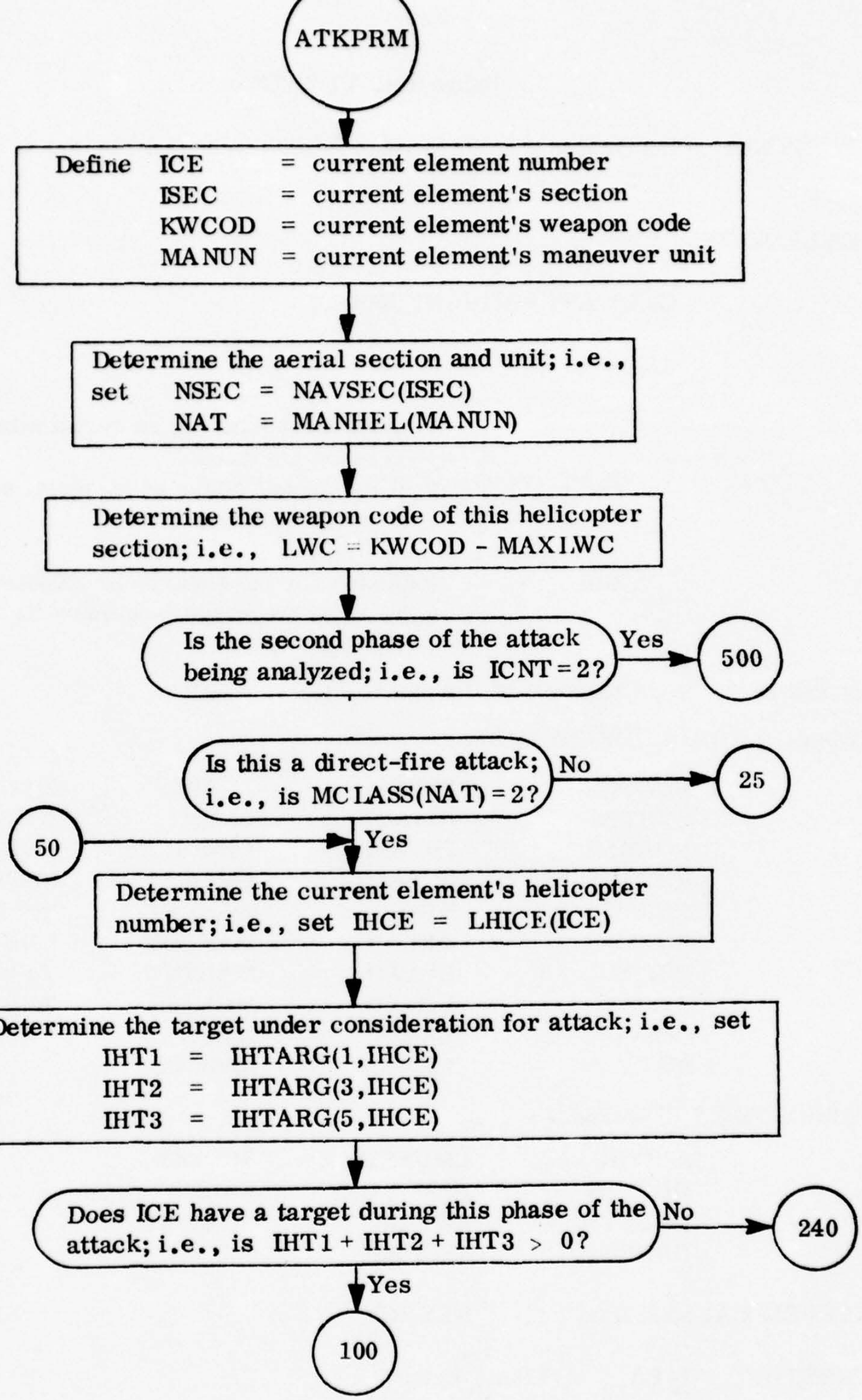
ECLOCK	ISORG	LNFIR	NTELE
EVHTIM	IUNACT	LNUM	NUMBER
FORMSE	JPHASE	LRNDC	NVOLM
ICECOM	JUNACT	LTARG	SEQPAR
IFBMIS	KFO	MANHEL	SHBM
IHAMO	KFRND	MANLDR	TDFRDY
IHDFMC	LFLAG	MANTYP	TFLY
IHTARG	LFRND	MCLASS	THBM
IPHASE	LHICE	MDFAF	
ISACT	LMUFL	NAVSEC	

SUBROUTINES REQUIRED

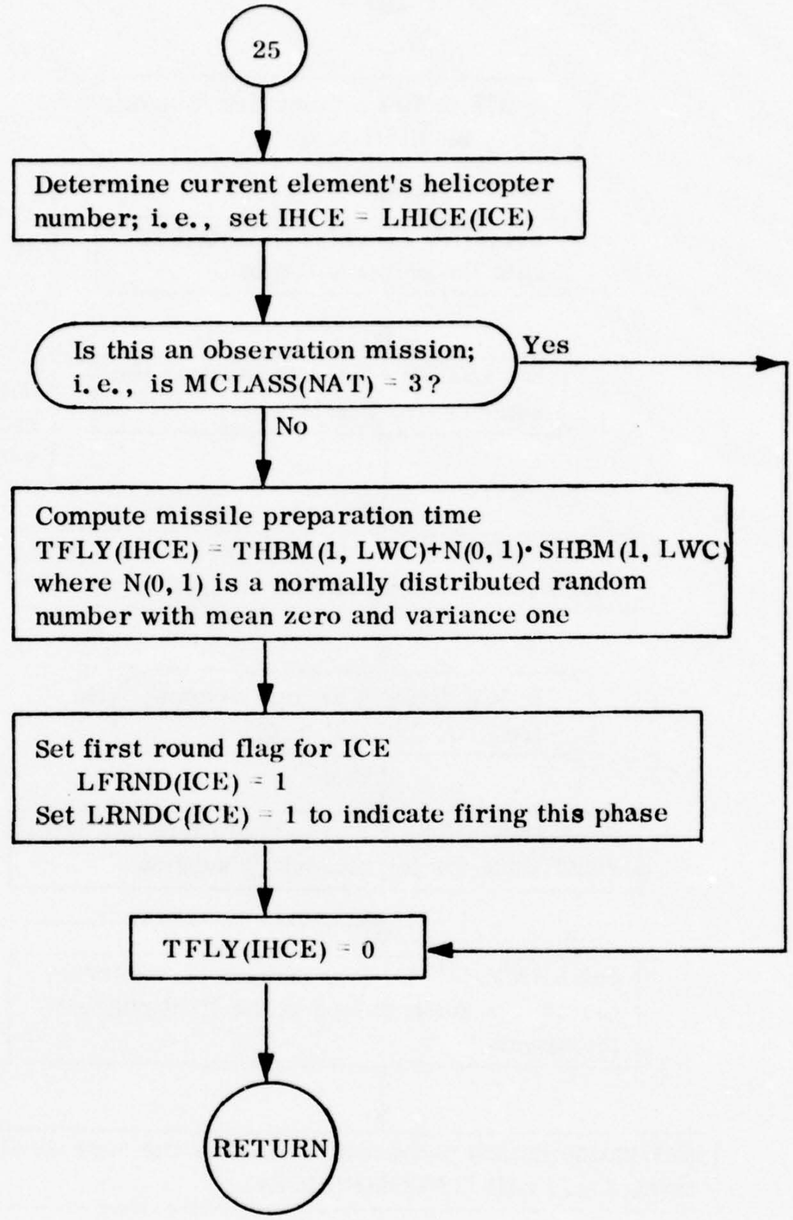
AIRFIR	LMUSET	SECSET
DEFPOS	PRMSET	SNRC
DELETH	RETIRE	TFCOMP
HFORM	SECPRM	

ATKPRM CALLED BY: HELFIR

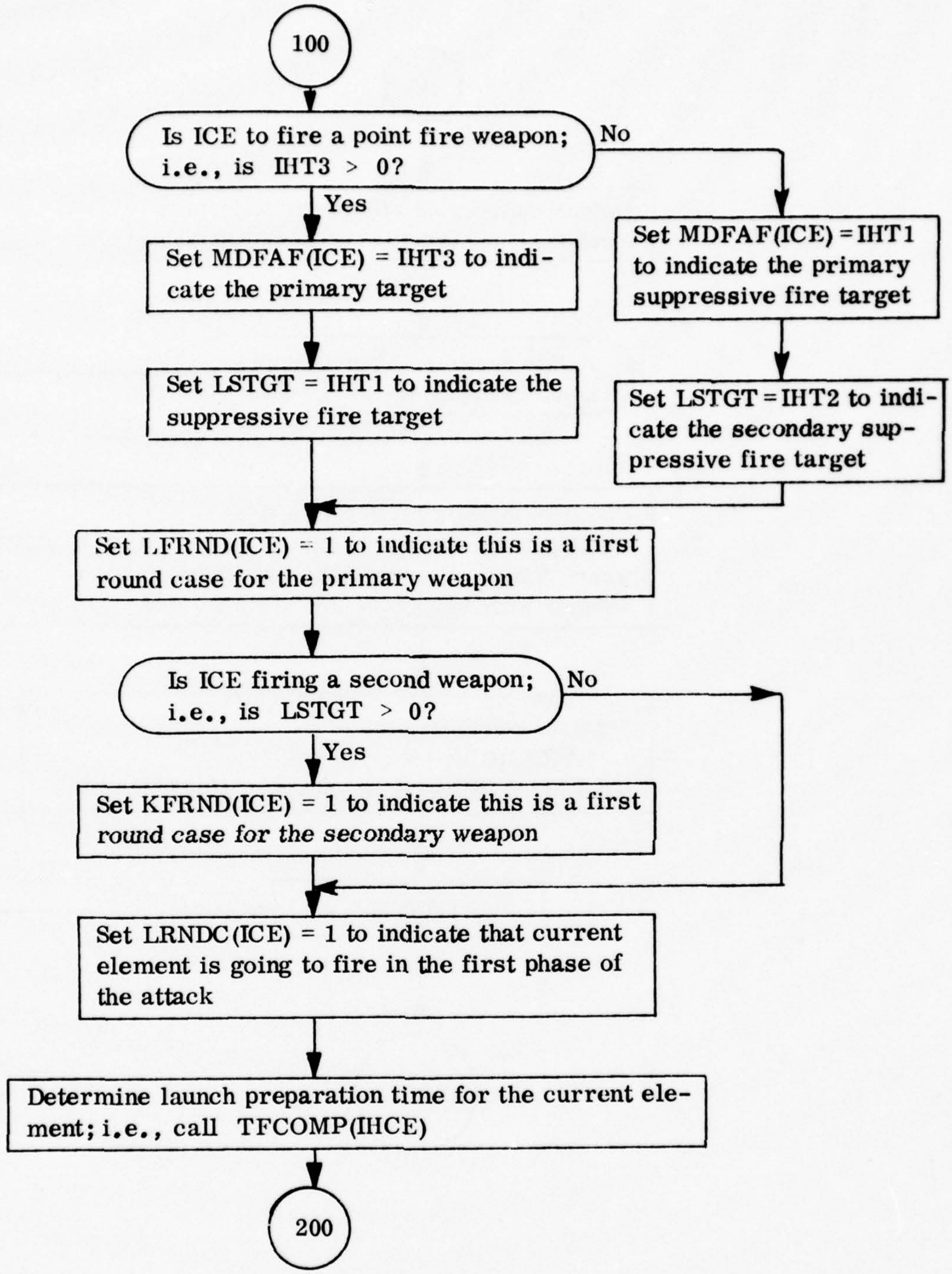
LENGTH: $14EA_{16}$ - 5354_{10} bytes



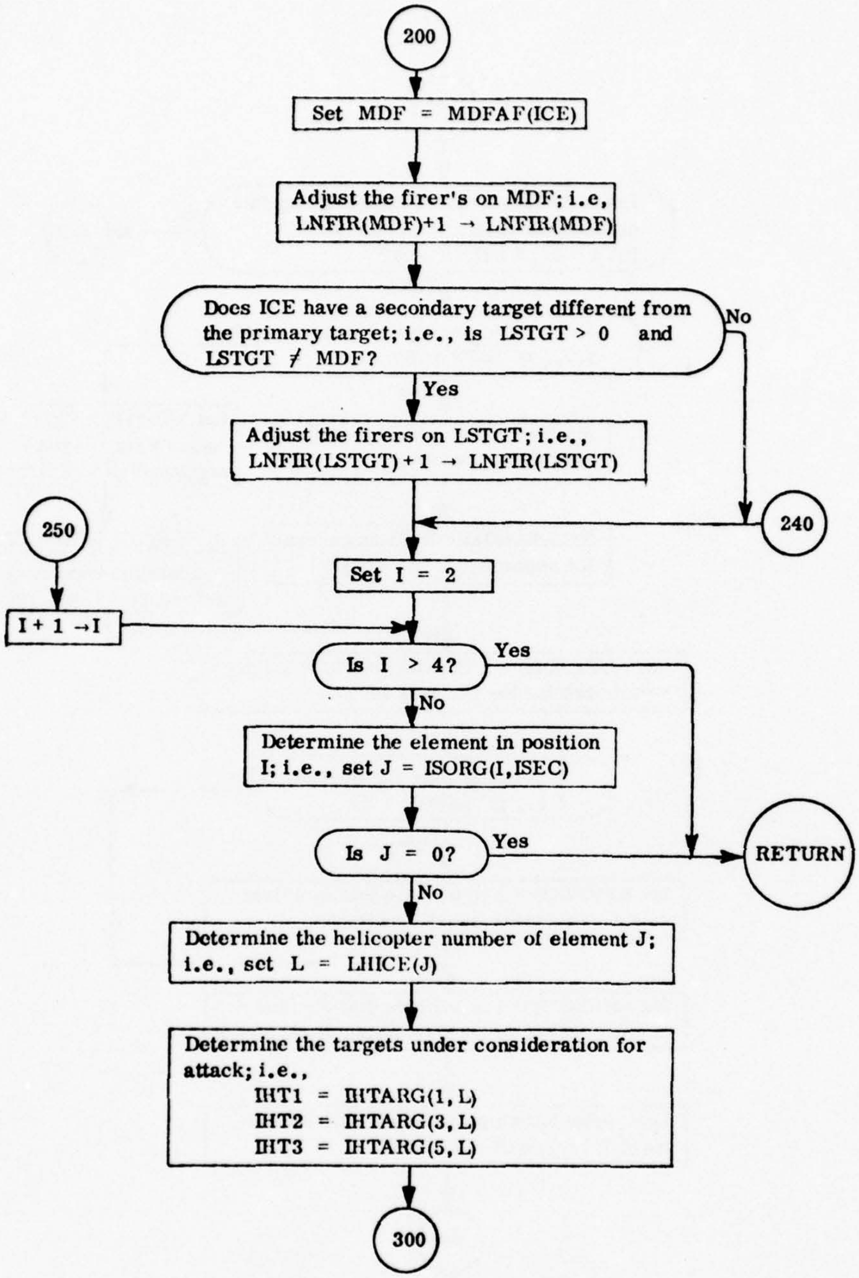
Subroutine ATKPRM



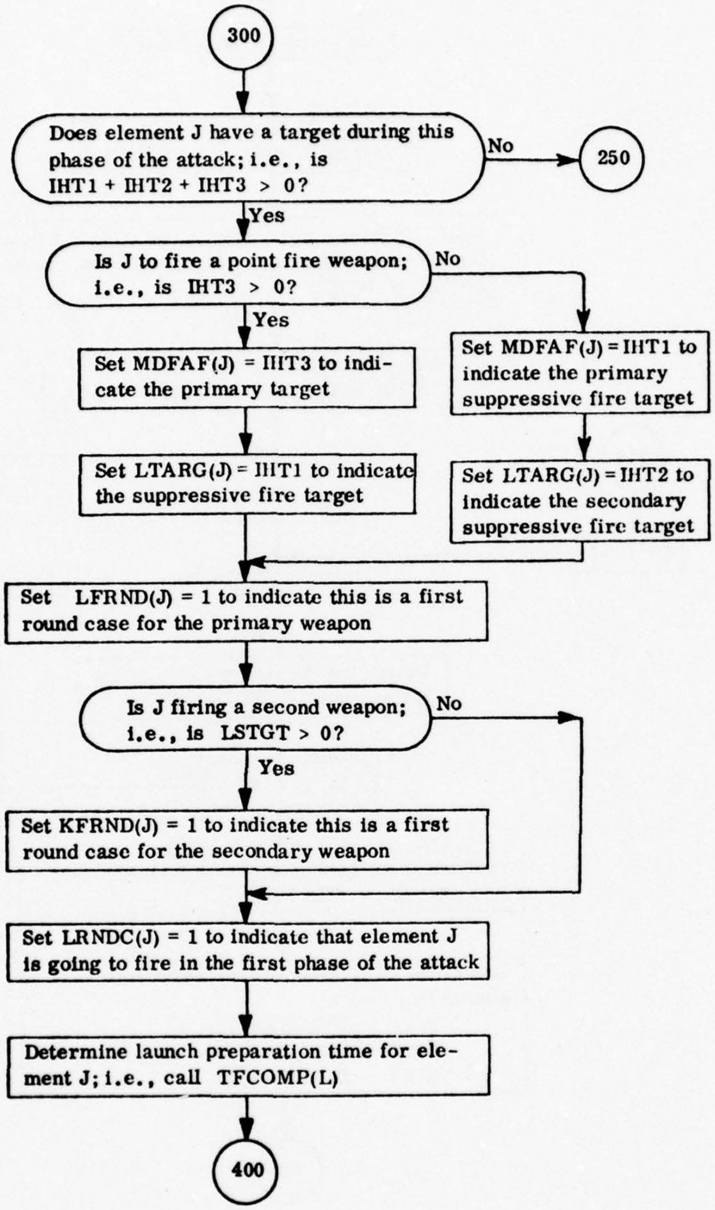
Subroutine ATKPRM: Continued



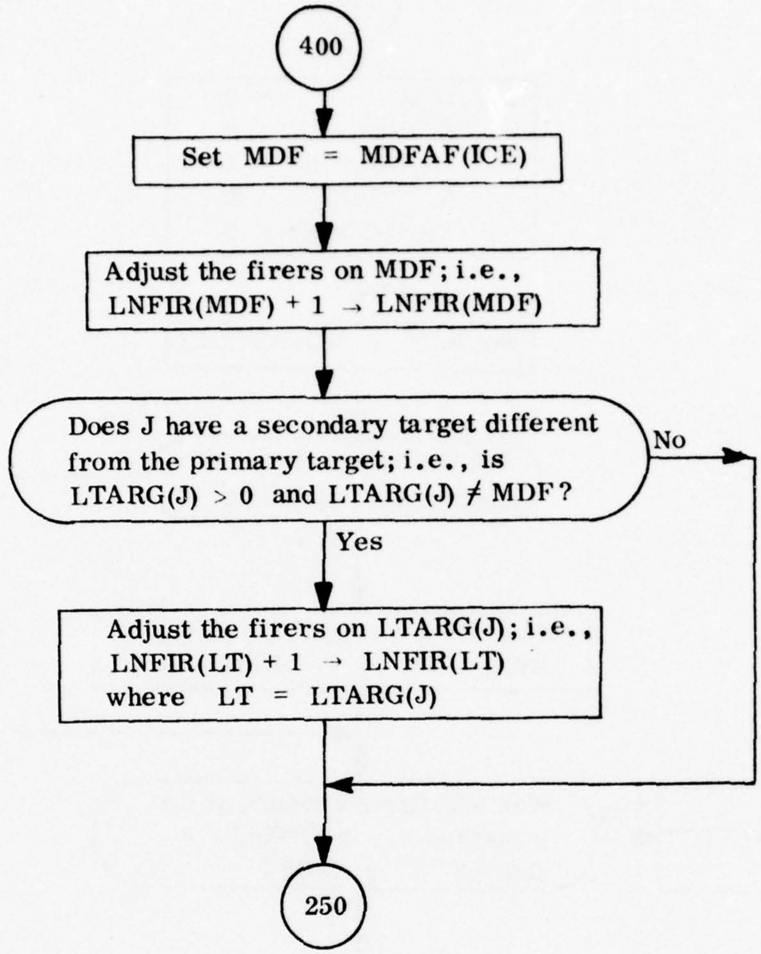
Subroutine ATKPRM: Continued



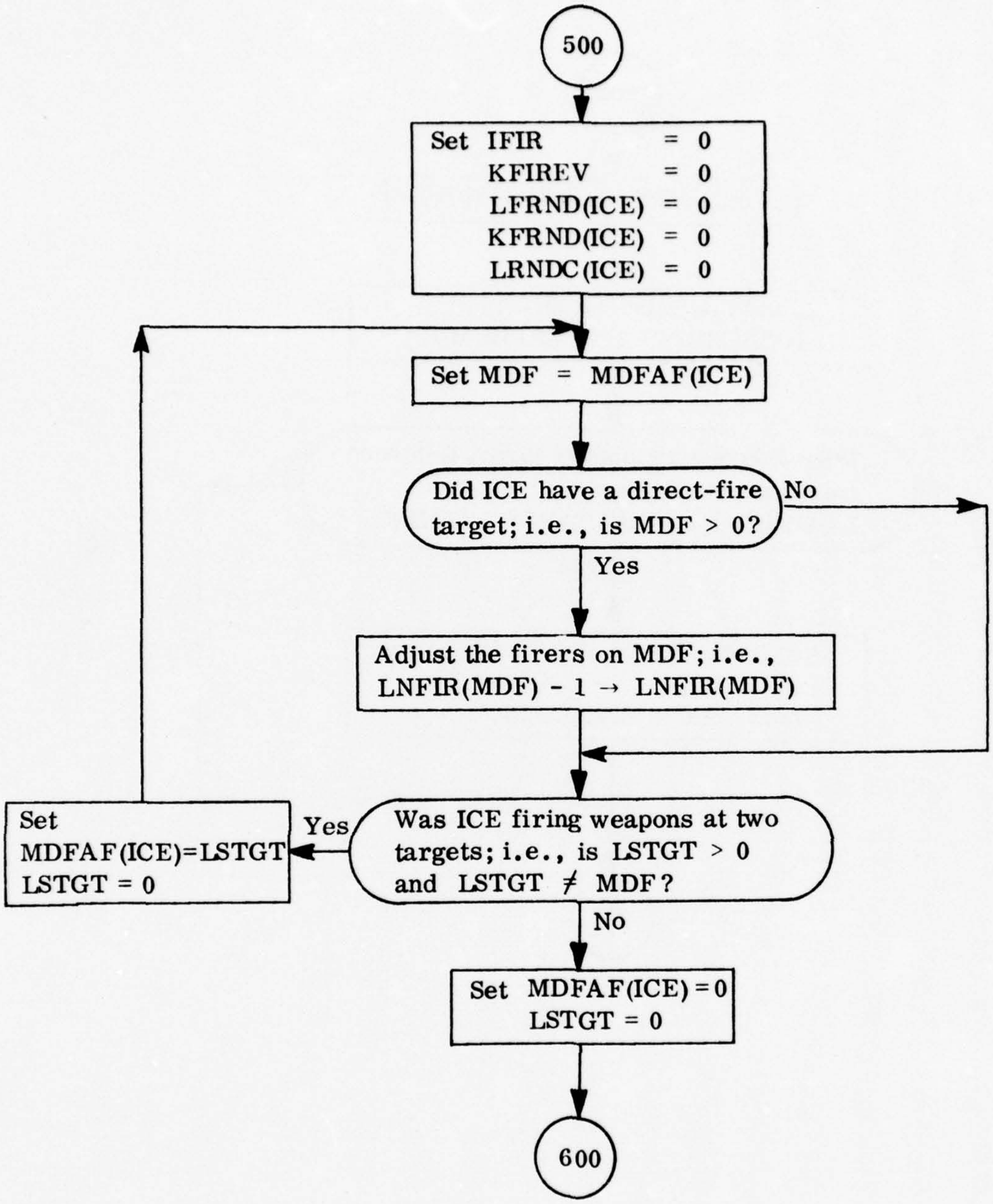
Subroutine ATKPRM: Continued



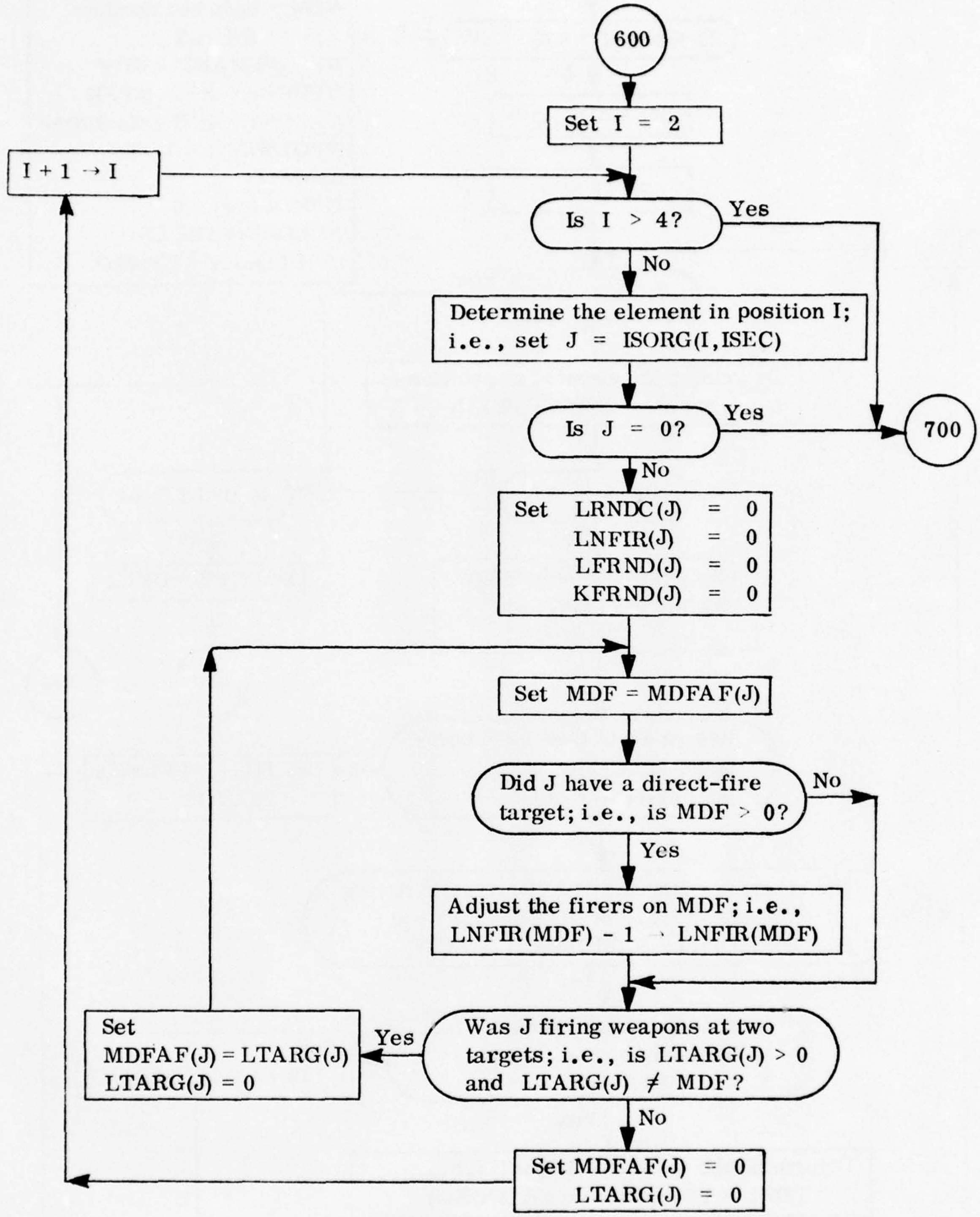
Subroutine ATKPRM: Continued



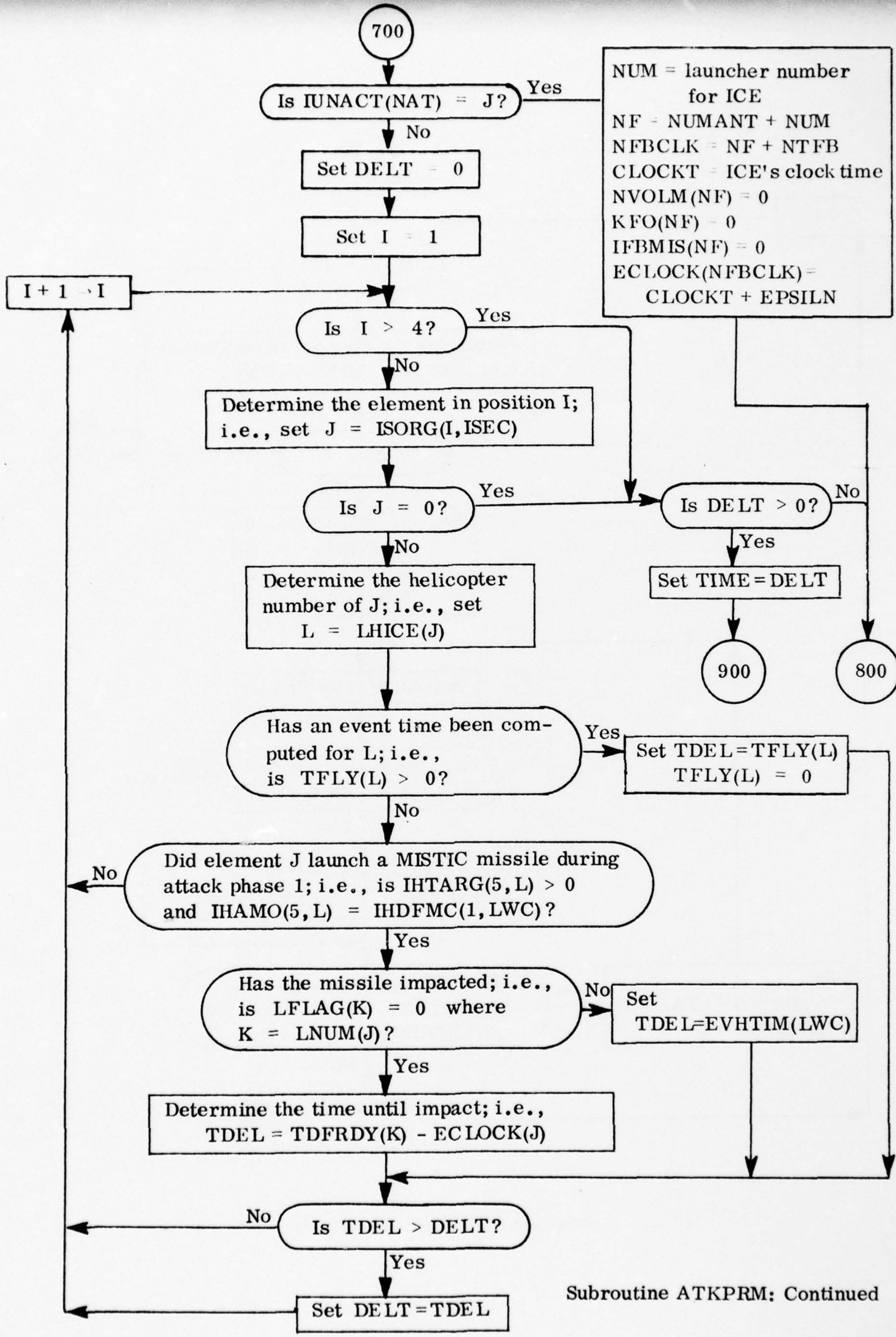
Subroutine ATKPRM: Continued



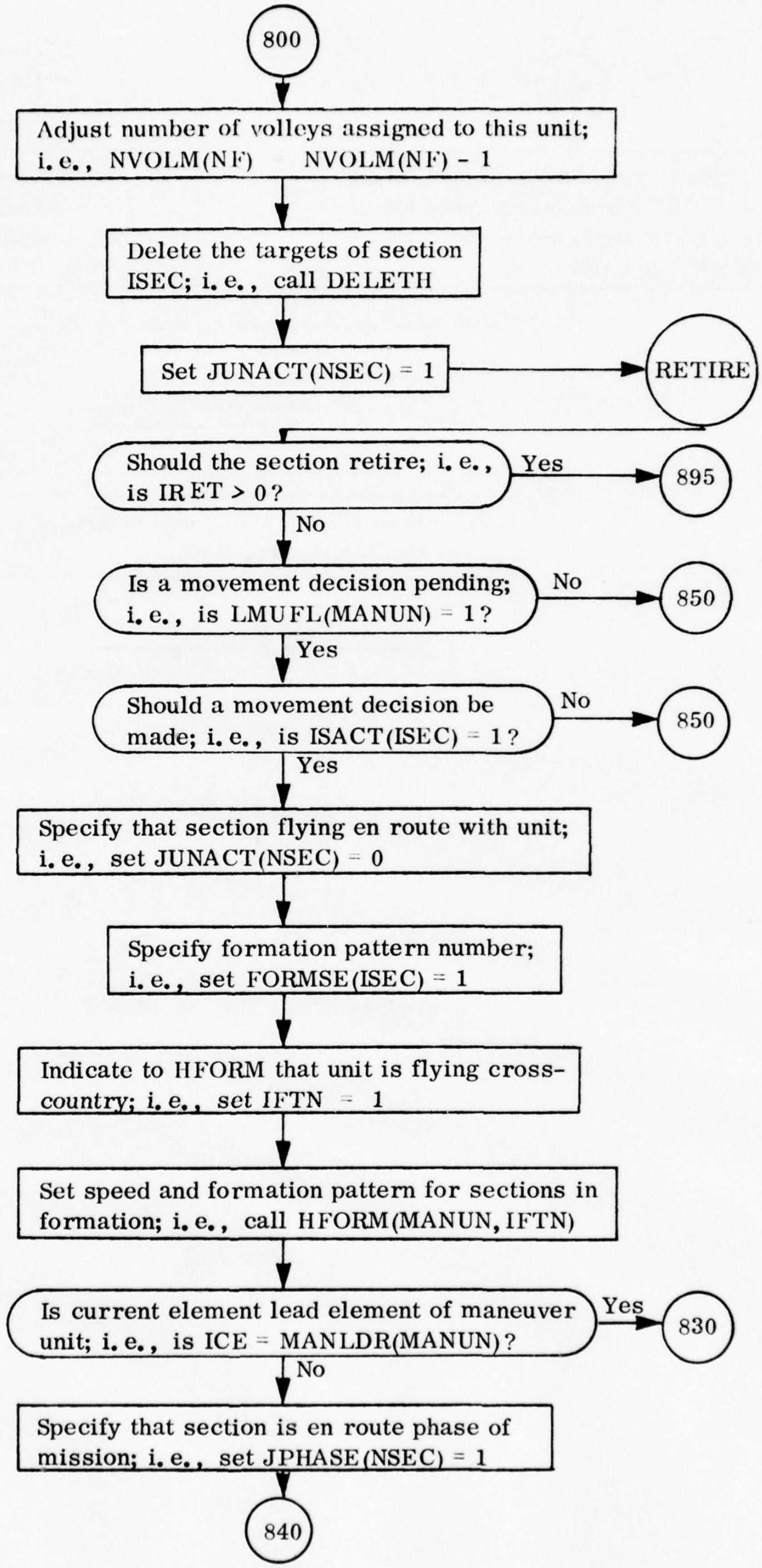
Subroutine ATKPRM: Continued



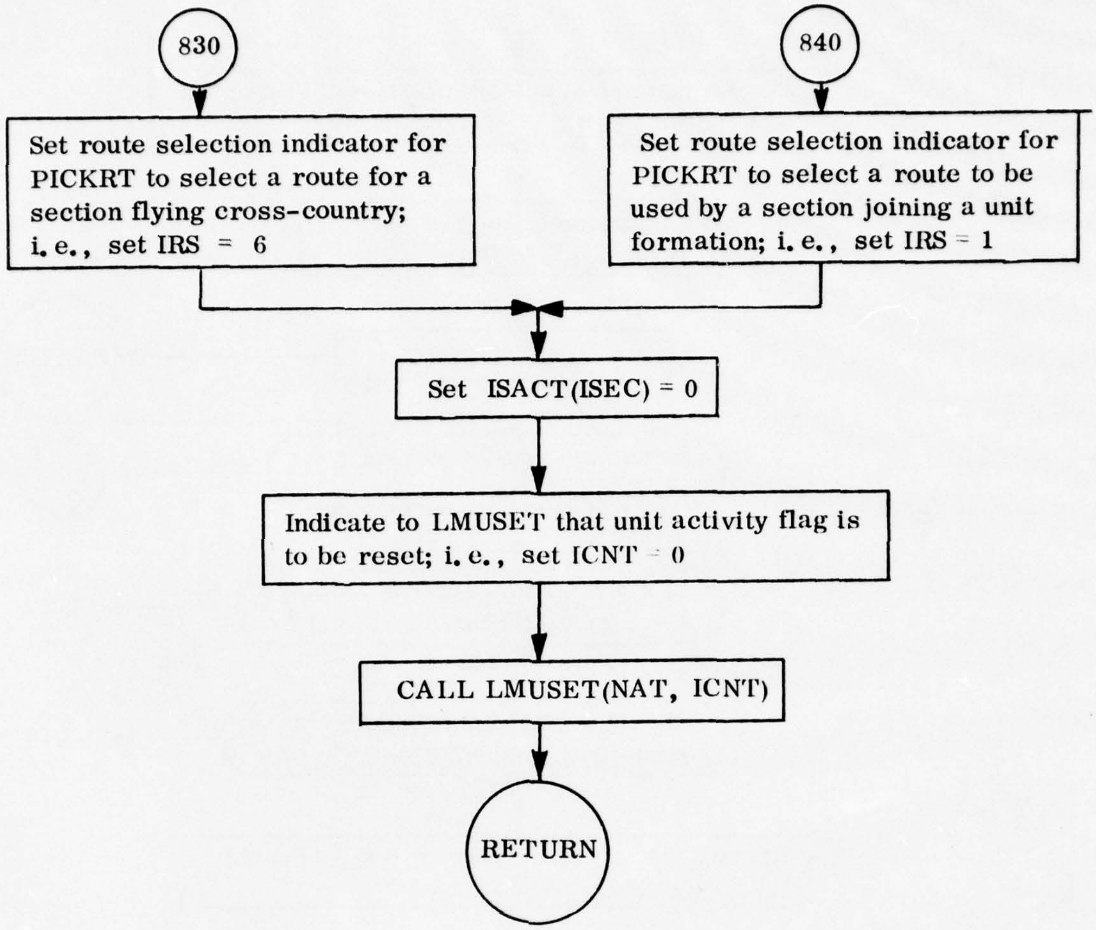
Subroutine ATKPRM: Continued



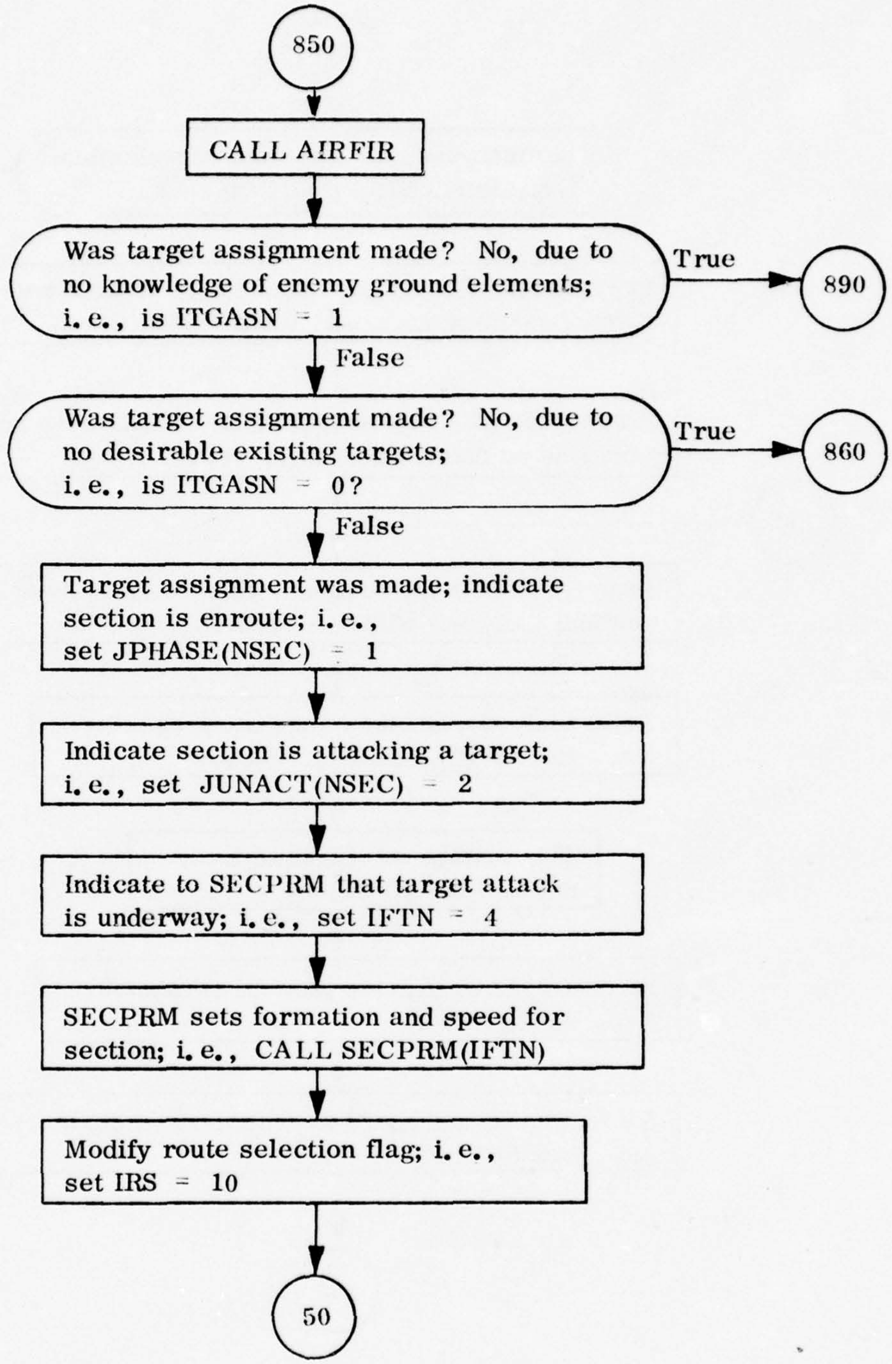
Subroutine ATKPRM: Continued



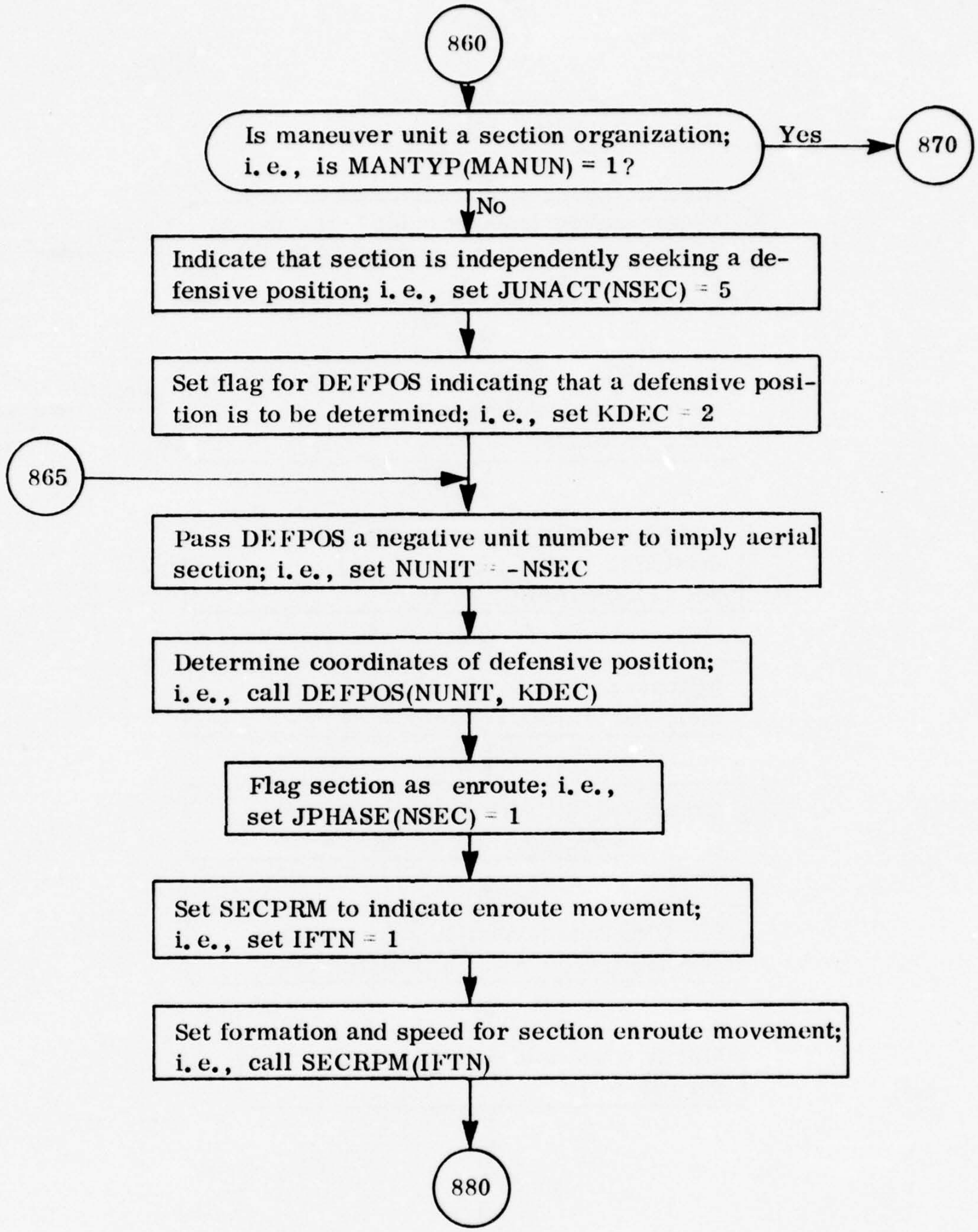
Subroutine ATKPRM: Continued



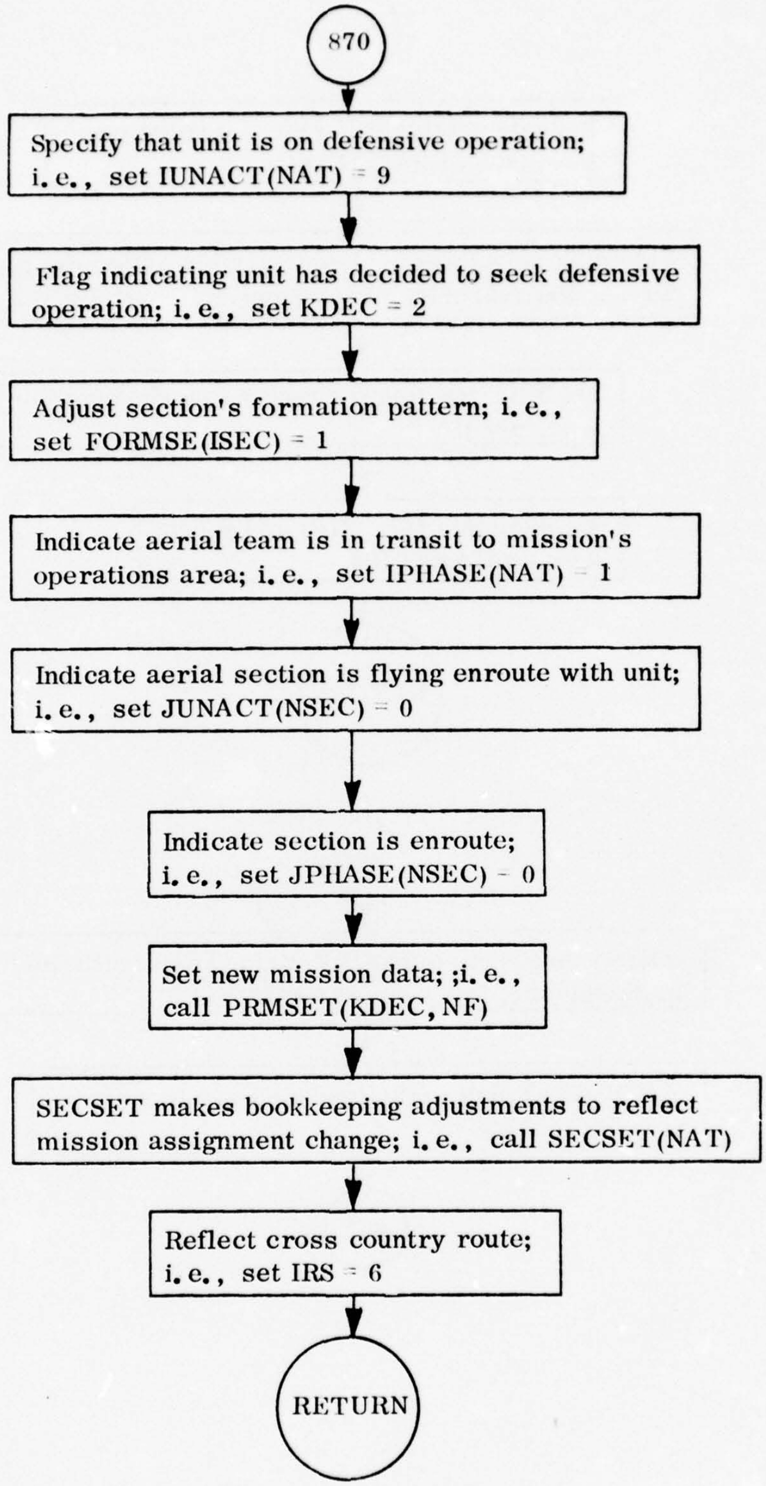
Subroutine ATKPRM: Continued



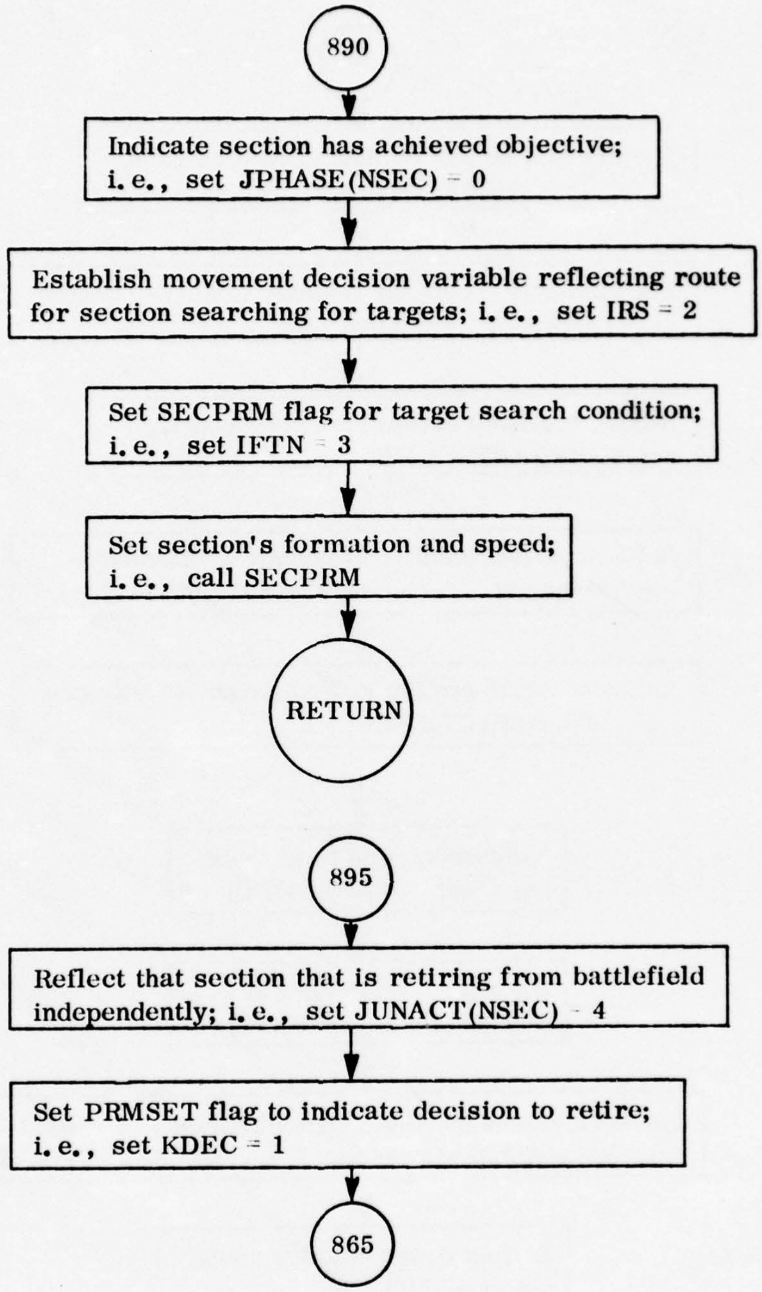
Subroutine ATKPRM: Continued

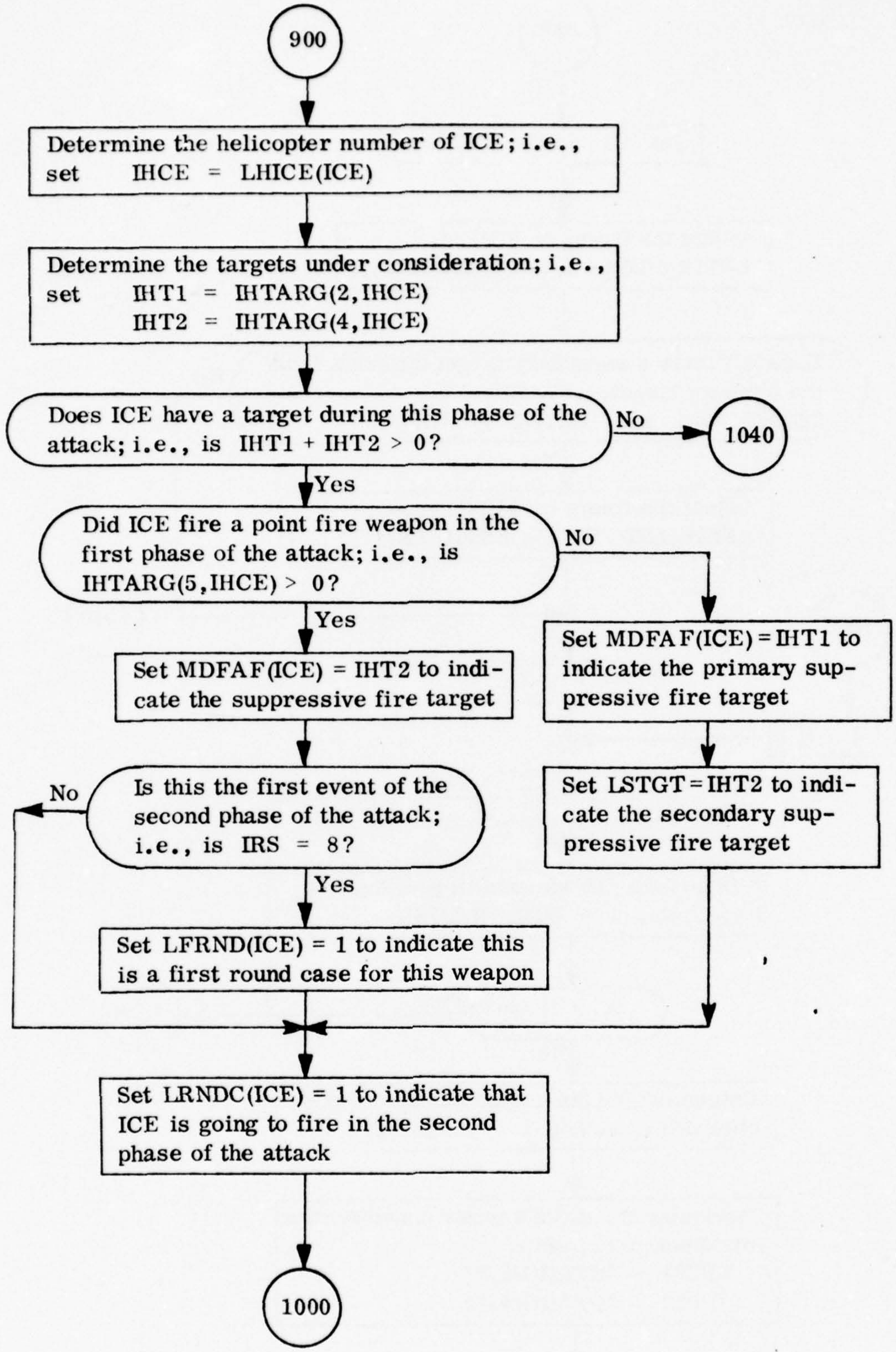


Subroutine ATKPRM: Continued

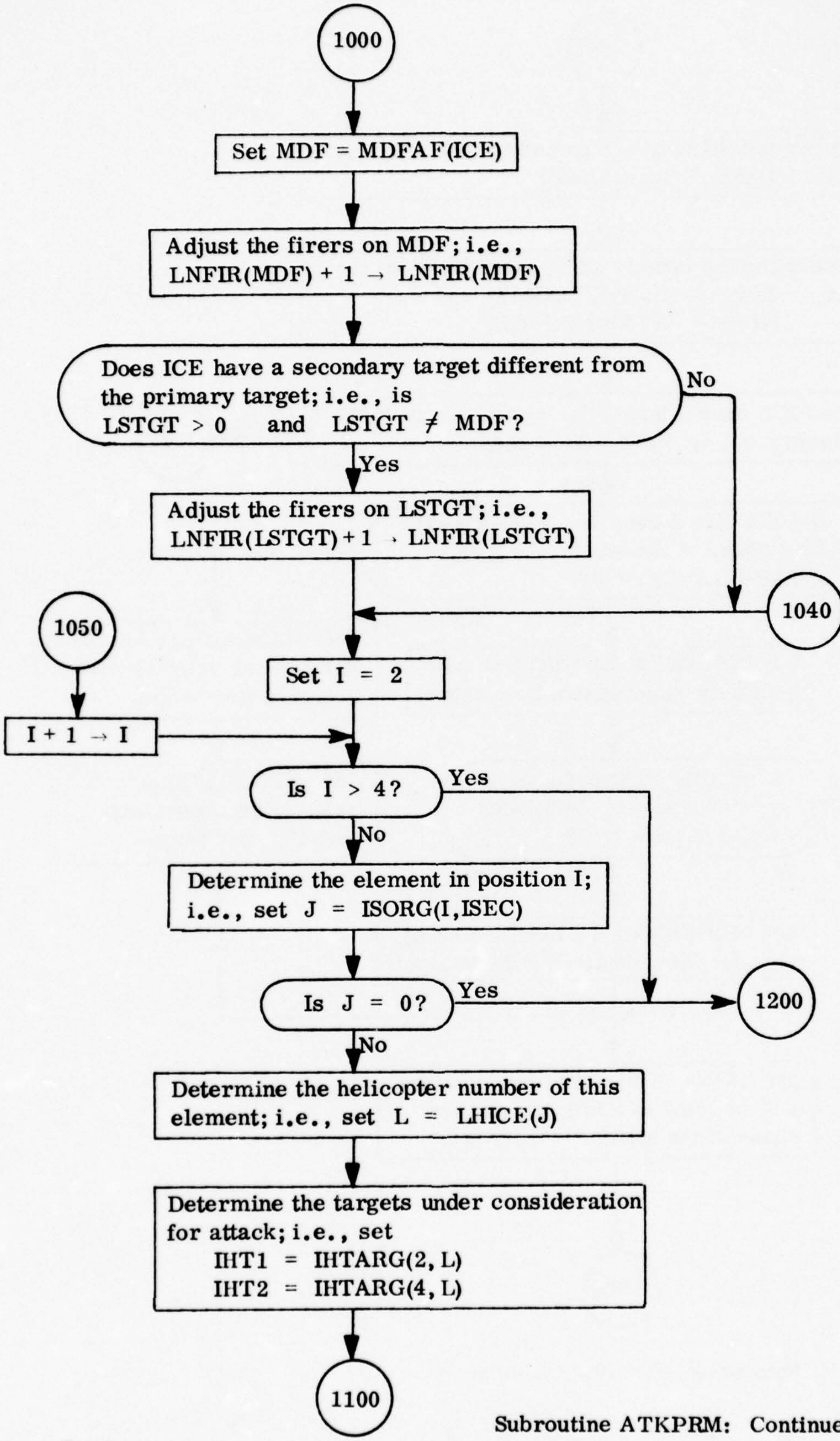


Subroutine ATKPRM: Continued

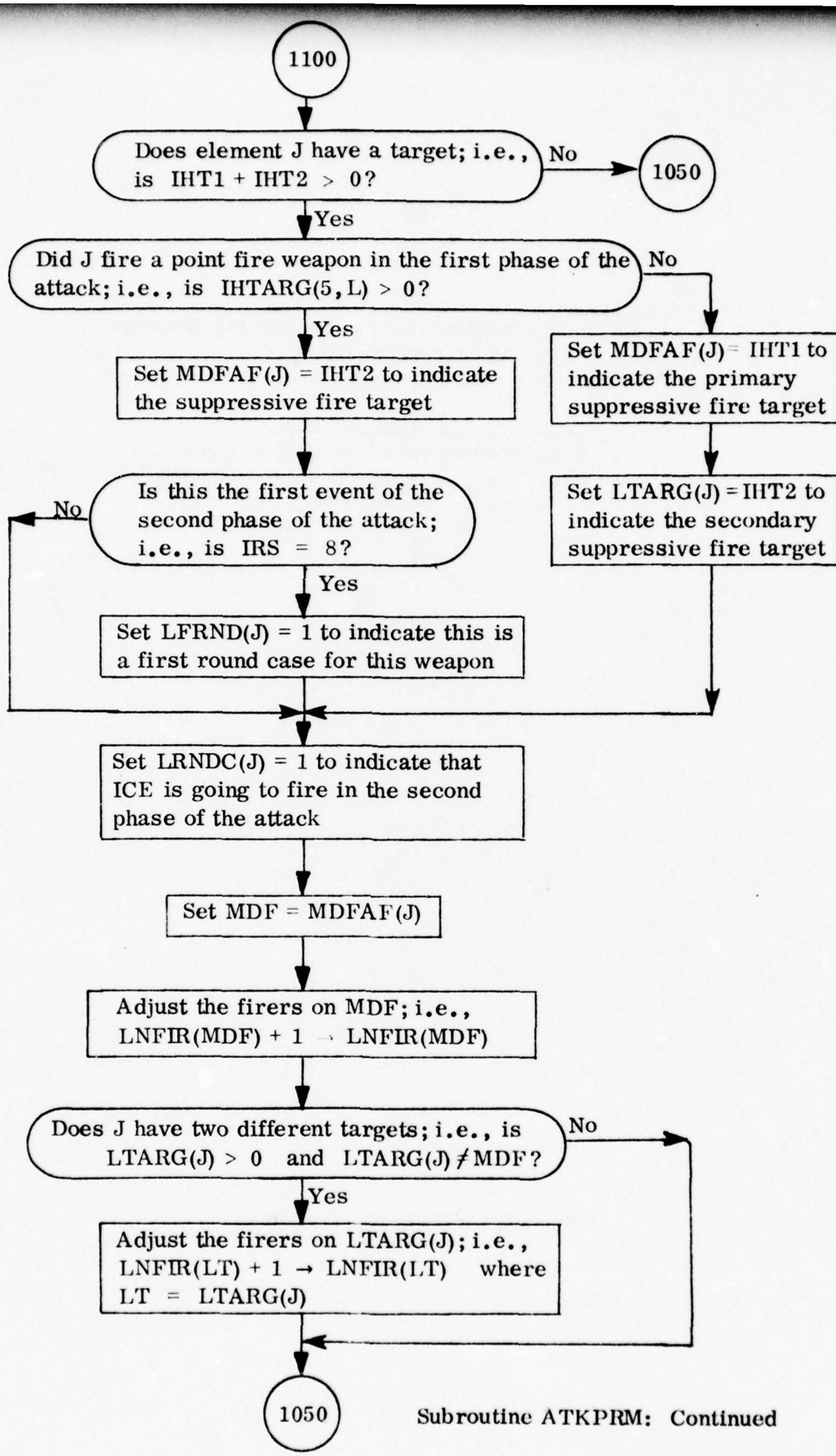

890
Indicate section has achieved objective; i.e., set JPHASE(NSEC) = 0
Establish movement decision variable reflecting route for section searching for targets; i.e., set IRS = 2
Set SECPRM flag for target search condition; i.e., set IFTN = 3
Set section's formation and speed; i.e., call SECPRM
RETURN
895
Reflect that section that is retiring from battlefield independently; i.e., set JUNACT(NSEC) = 4
Set PRMSET flag to indicate decision to retire; i.e., set KDEC = 1
865



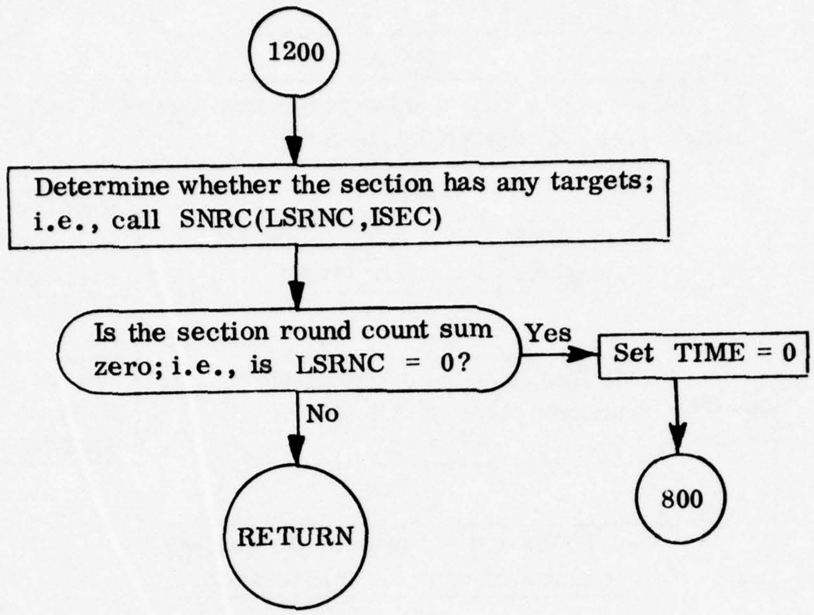
Subroutine ATKPRM: Continued



Subroutine ATKPRM: Continued



Subroutine ATKPRM: Continued



Subroutine ATKPRM: Continued

Subroutine AVAIL

PURPOSE: Subroutine AVAIL is called to determine a list of fire-support units available for mission assignment.

CALLING SEQUENCE:

```
CALL AVAIL(XC, YC, KLASS,KL, KU, IL2, LIST3, LIST4,
           LIST5, ISIG)
```

where

XC, YC = battlefield coordinates of requested mission target (input)

KLASS =
- 1 if artillery is the class of fire support being considered (input)
- 2 if MISTIC is the class of fire support being considered (input)
- 3 if aviation is the class of fire support being considered (input)

KL = first unit number to be examined for availability (input)

KU = last unit number to be examined for availability (input)

IL2 = number of fire-support units found to be available (output)

LIST3 = array containing the available fire-support unit numbers (output)

LIST4 = array containing the weapon codes of units in LIST3 (output)

LIST5 = array containing priority of missions currently being performed by units in LIST3 (output)

ISIG = array containing information about the availability of weapon codes (entry I has a value of one if weapon code I appears in LIST4; a zero otherwise).

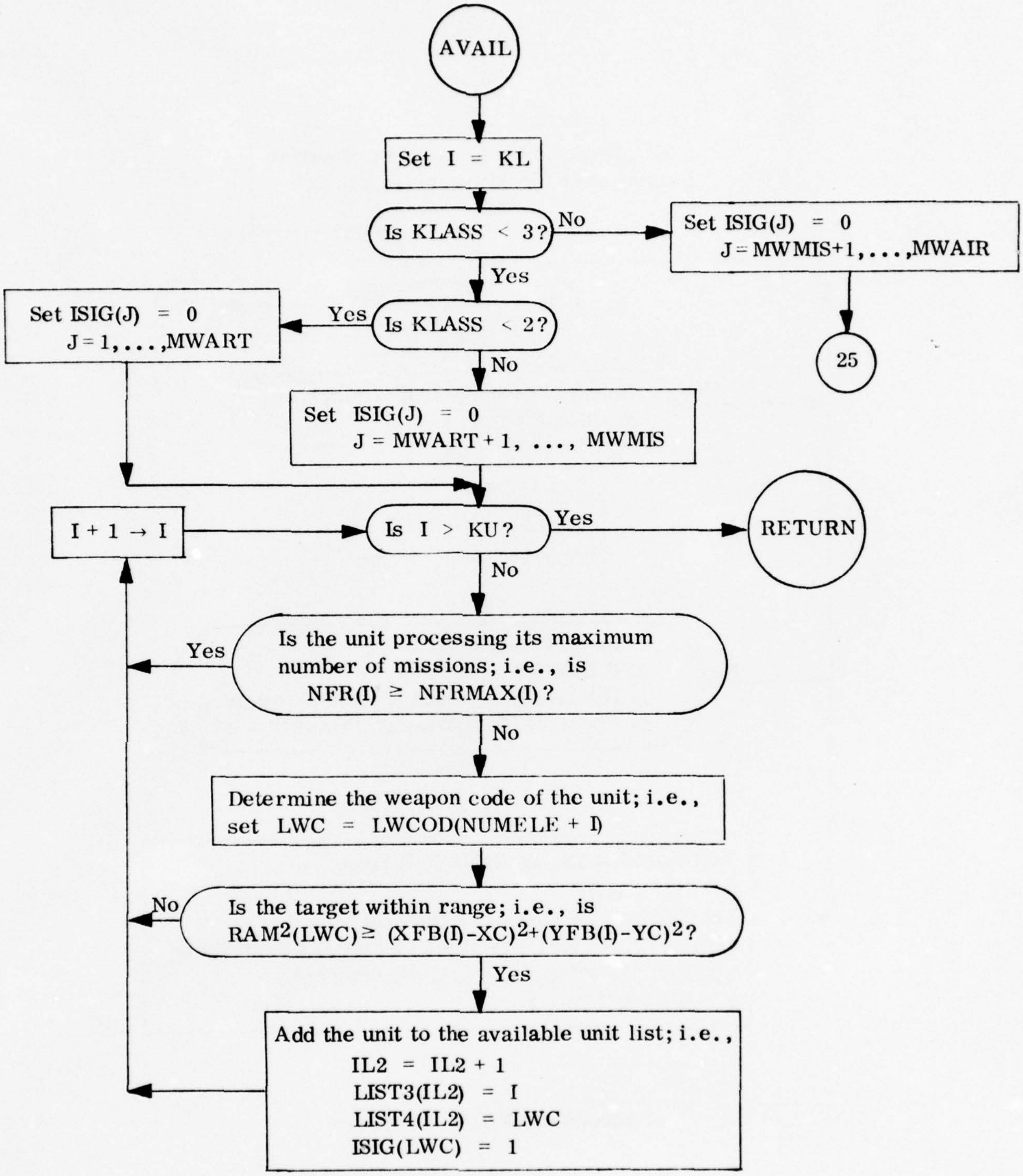
METHOD: Artillery and MISTIC units are available if their fire direction centers are not processing the maximum allowable number of fire requests. Aviation units are available if they are over the battlefield and are not engaged, or if they are waiting for mission assignment while off the battlefield. Only one class-of-fire support is considered per call to subroutine AVAIL.

COMMON AREAS REFERENCED:

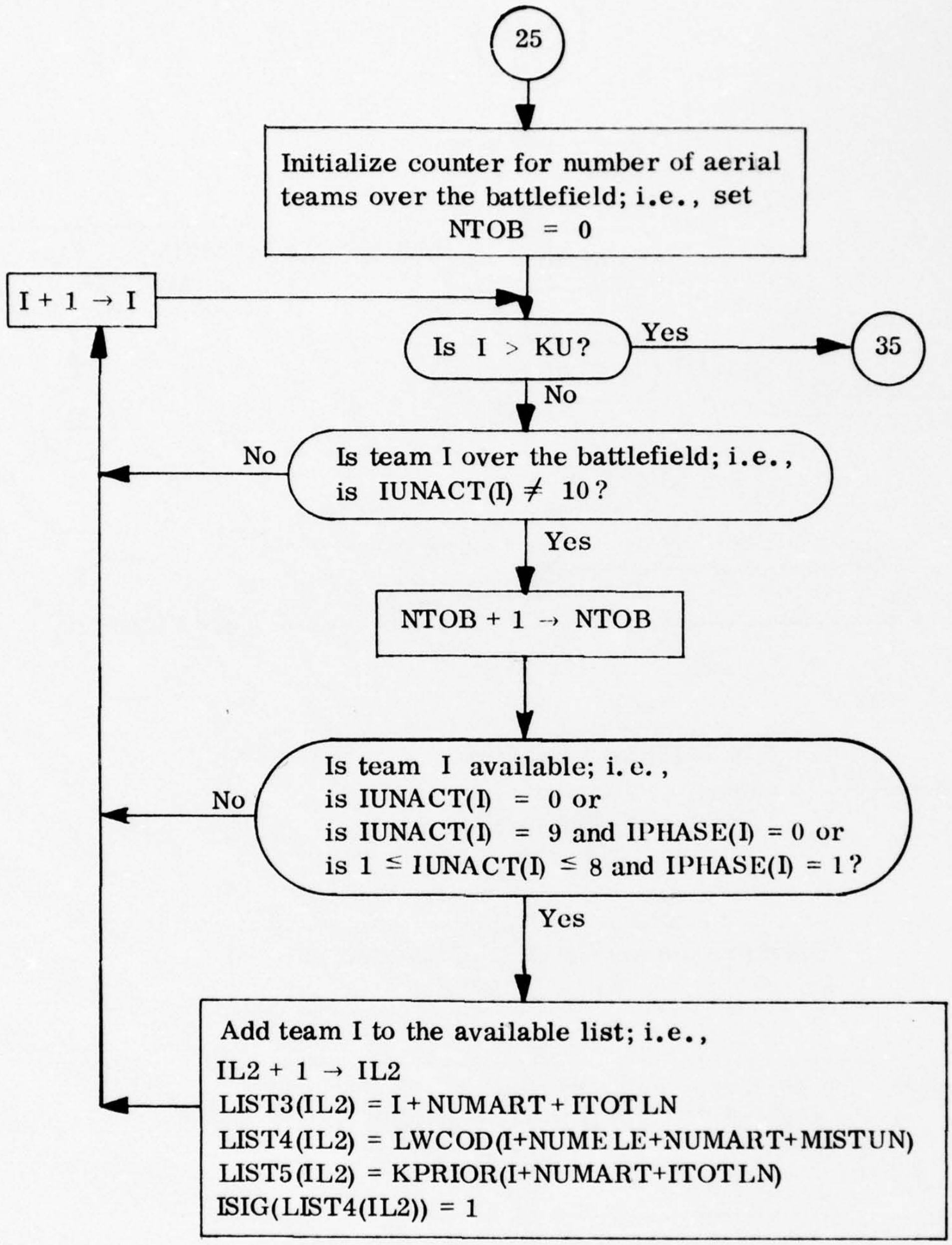
ICECOM	LWCOD	NUMBER
IPHASE	NFR	RAM
IUNACT	NFRMAX	XFB
KPRIOR	NTOBAL	YFB

SUBROUTINES REQUIRED:

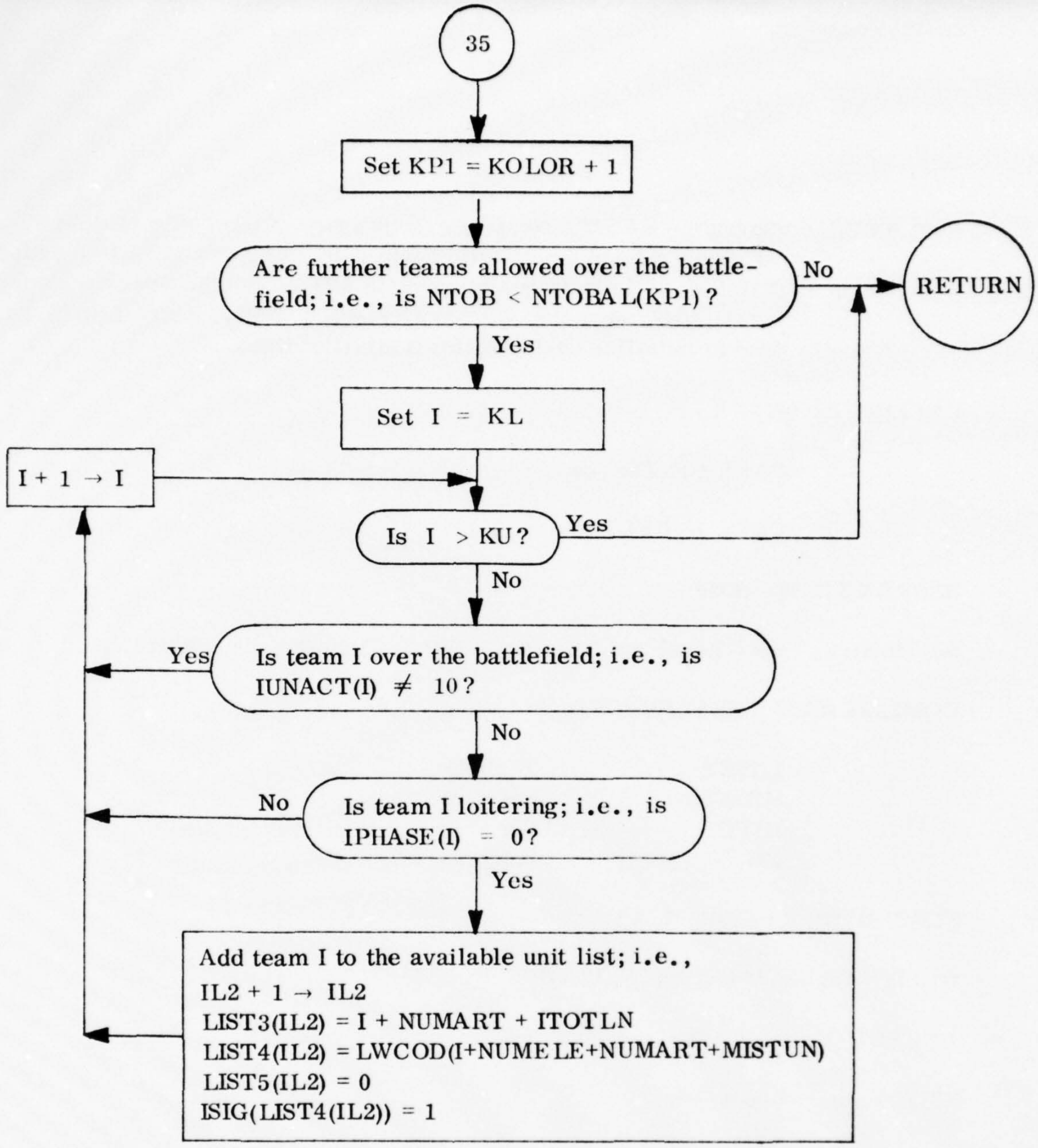
None



Subroutine AVAIL: Determining Available Fire Support Units



Subroutine AVAIL: Continued



Subroutine AVAIL: Continued

Subroutine BFLITE
Entry LNBFLT

PURPOSE: Subroutine BFLITE computes the position of a missile at time T, for either a level or ballistic flight path. The new missile location (XM, YM, XM1) is stored in common MIFO, and the down-range distance, RM1, is stored in common OPEN, RM1. Entry LNBFLT is used to initialize variables for a missile flight.

CALLING SEQUENCE:

CALL BFLITE or

CALL LNBFLT

RESTRICTIONS: None

METHOD: See "Level and Ballistic Flight," Chapter 2, in Volume 1.

COMMON AREAS REFERENCED:

LNSET	TYPMIS
MIDATA	VM
MIFO	ZM
RM	ZMD

SUBROUTINES USED: RGXY

BFLITE CALLED FROM: FLIGHT

LNBFLT CALLED FROM: LAUNCH

STORAGE: 506 words

BFLITE

Does the missile have a level flight profile, i.e., is TYPMIS(ILNNUM) < 3?

Yes / No

Missile has a ballistic flight profile

Set VM1 = missile launch velocity

VM1 = VM(ILNNUM)

ANGLCH = launch angle

COSANG = COS(NAGLCH)

T = elapsed time since launch

RM1 = down-range horizontal distance

RM1 = T · VM1 · COSANG

ZM1 = missile elevation

ZLP = launch point elevation

SM1 = $-4.9 \cdot T^2 + VM1 \cdot T \cdot SIN(ANGLCH) + ZLP$

TANANG = $TAN(ANGLCH) - 9.8 \cdot RM1/(VM1^2 \cdot COSANG^2)$

Compute the flight angle with respect to a horizontal plane

ANGMIS = $\tan^{-1}(TANANG)$

Missile has a level flight path

Z1 = a normally distributed random number obtained from subroutine FLIGHT

B3 = missile elevation at distance point J-1

B3 = ZM(J-1) + Z1 · ZMD(J-1)

Compute down-range distance and altitude at time T

RM1 = B1 · (RM(J) - RM(J-1) + RM(J-1)

ZM1 = B1 · (SM(J) + Z1 · ZMD(J) - B3) + B3 + ZLP

(XLP, YLP) = launch point coordinates

COSM = cosine of missile flight direction

SINM = sine of missile flight direction

Compute missile battlefield coordinates

XM = XLP + RM1 · COSM

YM = YLP + RM1 · SINM

RETURN

Subroutine BFLITE: Missile Position at Time T

LNBFLT

VM1 = missile launch velocity
VM1 = VM(ILNNUM)
RMO = horizontal distance from launch point to target coordinates
RMO = RGXY(XLP, YLP, XAT, YAT)

Does the missile have a level flight profile, i.e., is TYPMIS(ILNNUM) < 3? — Yes → 20

No

Missile has a ballistic trajectory
RMX = maximum possible flight distance
RMX = $VM1^2/9.8$

Can the missile reach the target, i.e., is RMX ≥ RMO? — Yes → 15

No

Missile cannot reach target
ANGLCH = launch angle
ANGLCH = /4
Compute new target coordinates
XAT = XLP + (RMX/RMO)(XAT - XLP)
YAT = YLP + (RMX/RMO)(YAT - YLP)
Compute projected impact time
TIMP = TM + VM1 · .14431

RETURN

Subroutine BFLITE: Continued

15

Missile can reach the target
Compute the launch angle
ANGLCH = .5 · arcsin $(9.8 \cdot RMO/VM1^2)$
Compute the projected impact time
RIMP = TM + VM1 · sin(ANGLCH)/4.9

RETURN

20

Missile has a level flight path
Compute the projected impact time
TIMP = TM + RMO/VM1

RETURN

Subroutine BFLITE: Continued

Subroutine CASHEL

PURPOSE: Subroutine CASHEL is called after an air-defense firing event to convert lethality codes prepared by the air-defense model into physical damage data of use to the aerial vehicle operations models.

CALLING SEQUENCE:

CALL CASHEL(THIT,I)

where

THIT = battle time at which lethality is assessed, and

I = element number of damaged vehicle.

METHOD: See Chapter 9 of Volume 1.

COMMON AREAS REFERENCED:

FMOKIL	LTHCOD
KILFIR	LWCOD
LAMMO	NUMBER
LHICE	TRET

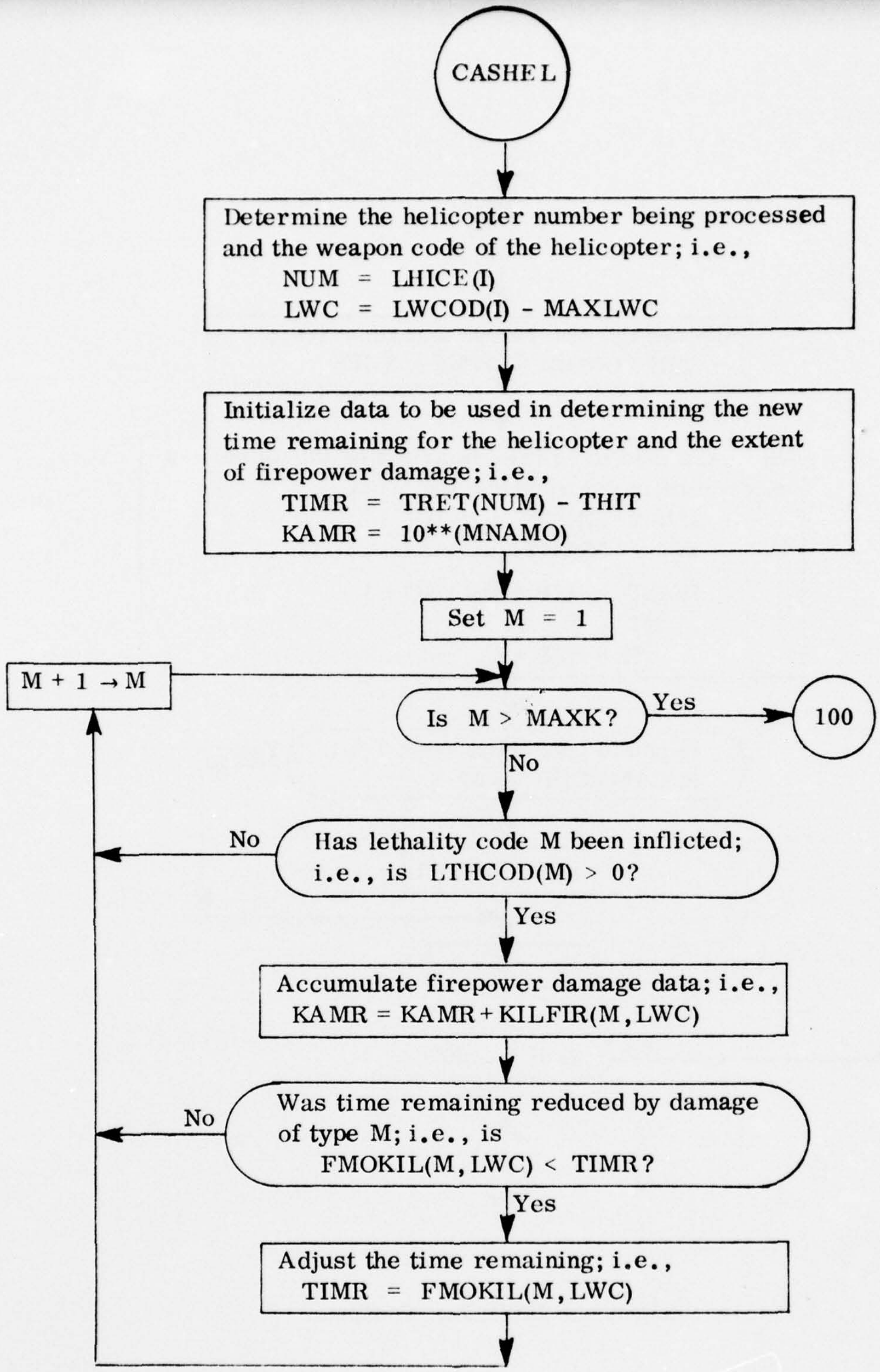
SUBROUTINES REQUIRED:

None

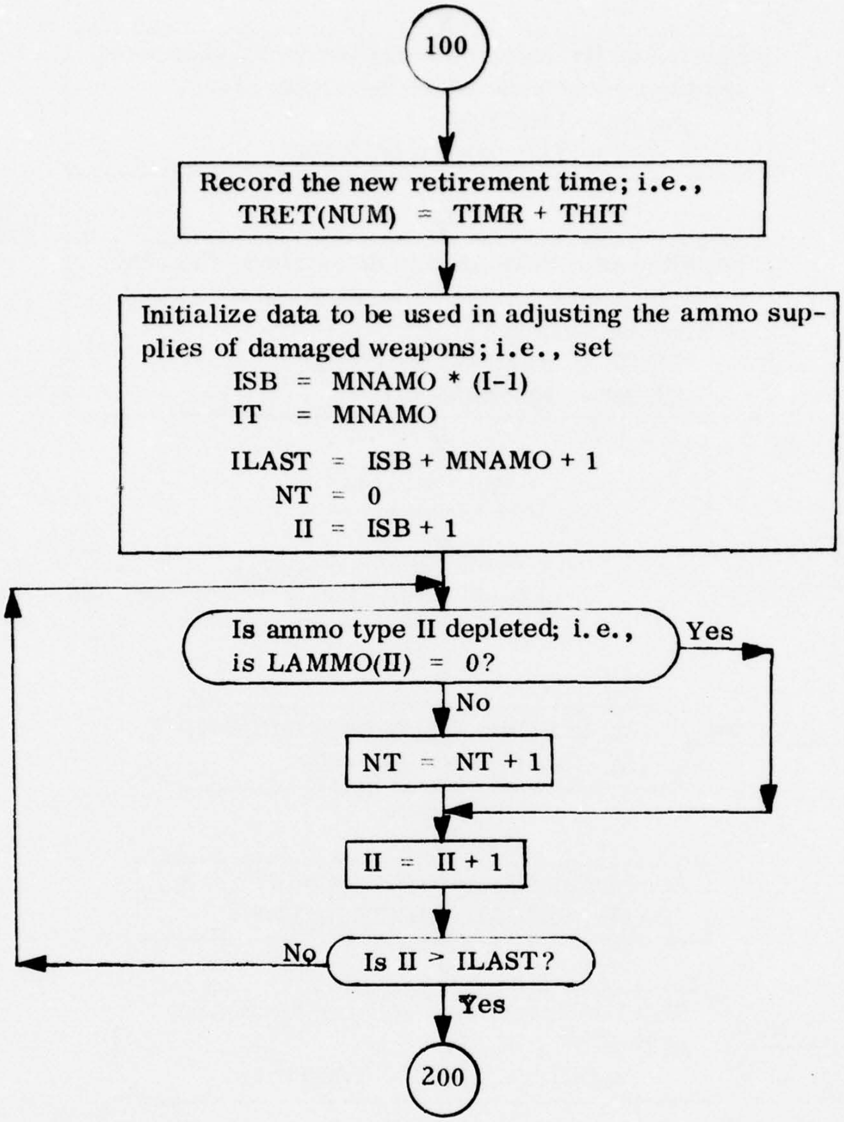
CASHEL CALLED BY:

ADGCON

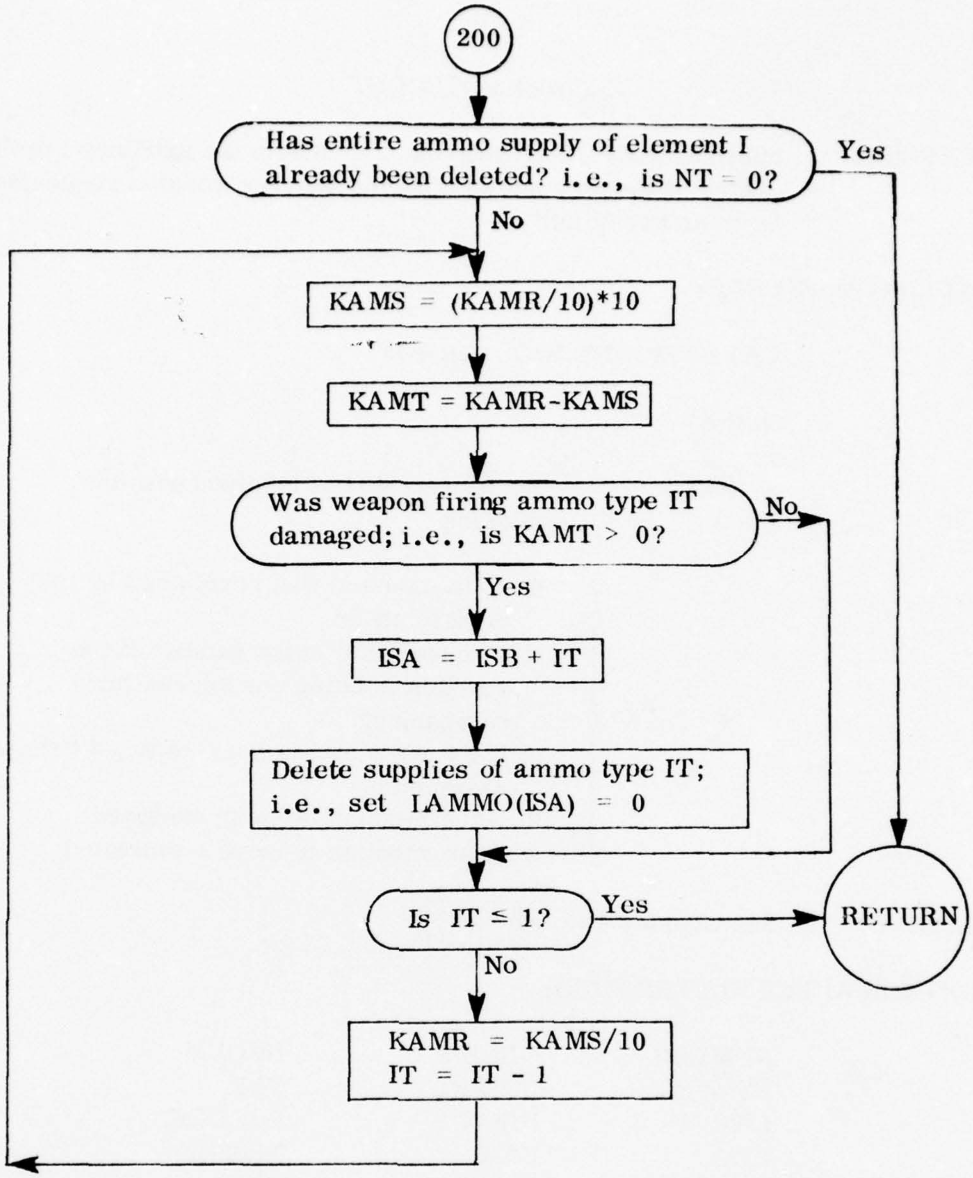
LENGTH: $564_{16} = 1380_{10}$ bytes



Subroutine CASHEL: Lethality Code Conversion



Subroutine CASHEL: Continued



Subroutine CASHEL: Continued

Subroutine CBCONT

PURPOSE: Subroutine CBCONT sets flags to inform the artillery intelligence center of the outcome of a counterbattery mission requested from an aerial unit.

CALLING SEQUENCE:

CALL CBCONT(NAT,N,ICNT)

where

NAT = aerial unit that is involved with the mission

N =
- 0 if the mission was performed by the aerial unit or
- fire request list entry number if the mission is being considered for assignment
- 1 if the mission has been rejected before assignment
- 2 if the mission is being assigned
- 3 if the mission is being terminated

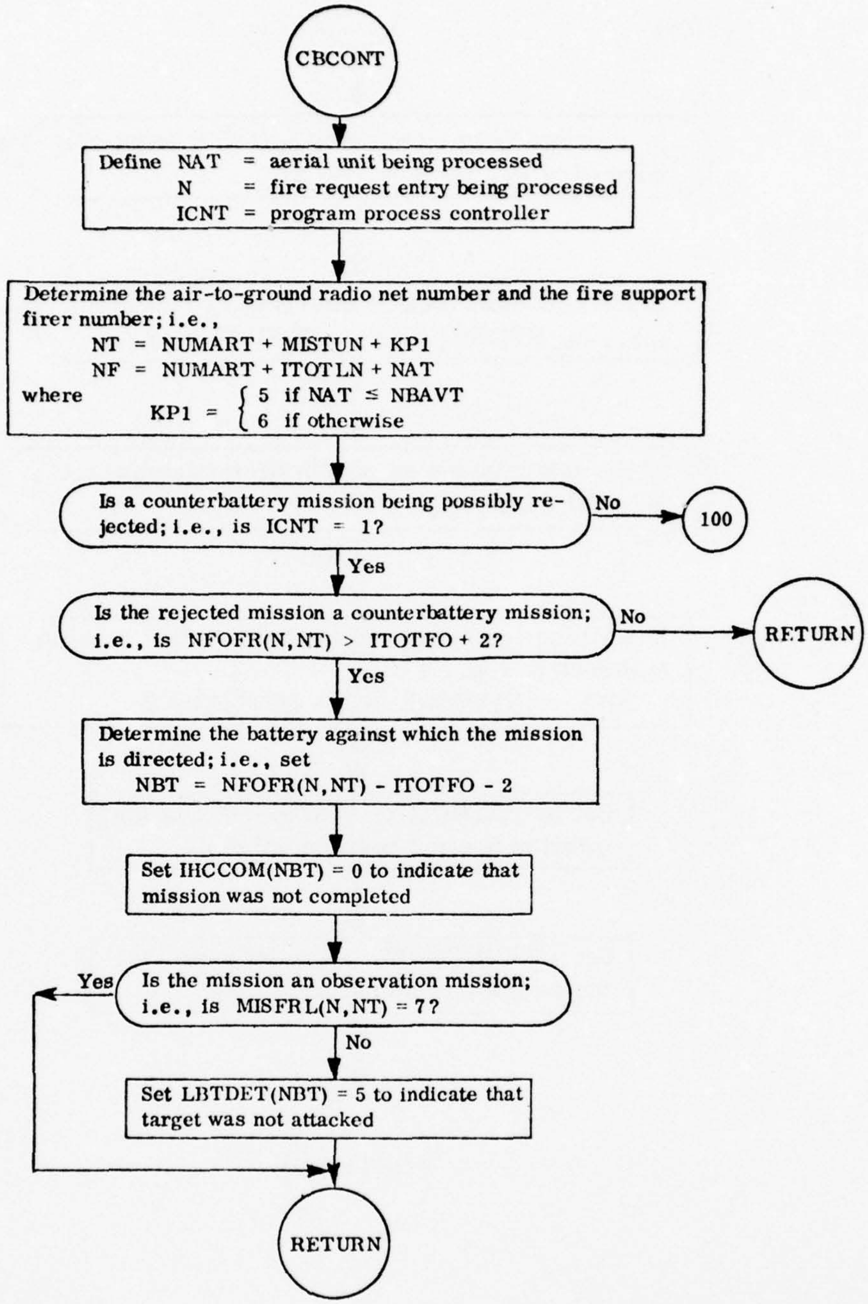
METHOD: See Chapter 4 of Volume 1.

COMMON AREAS REFERENCED:

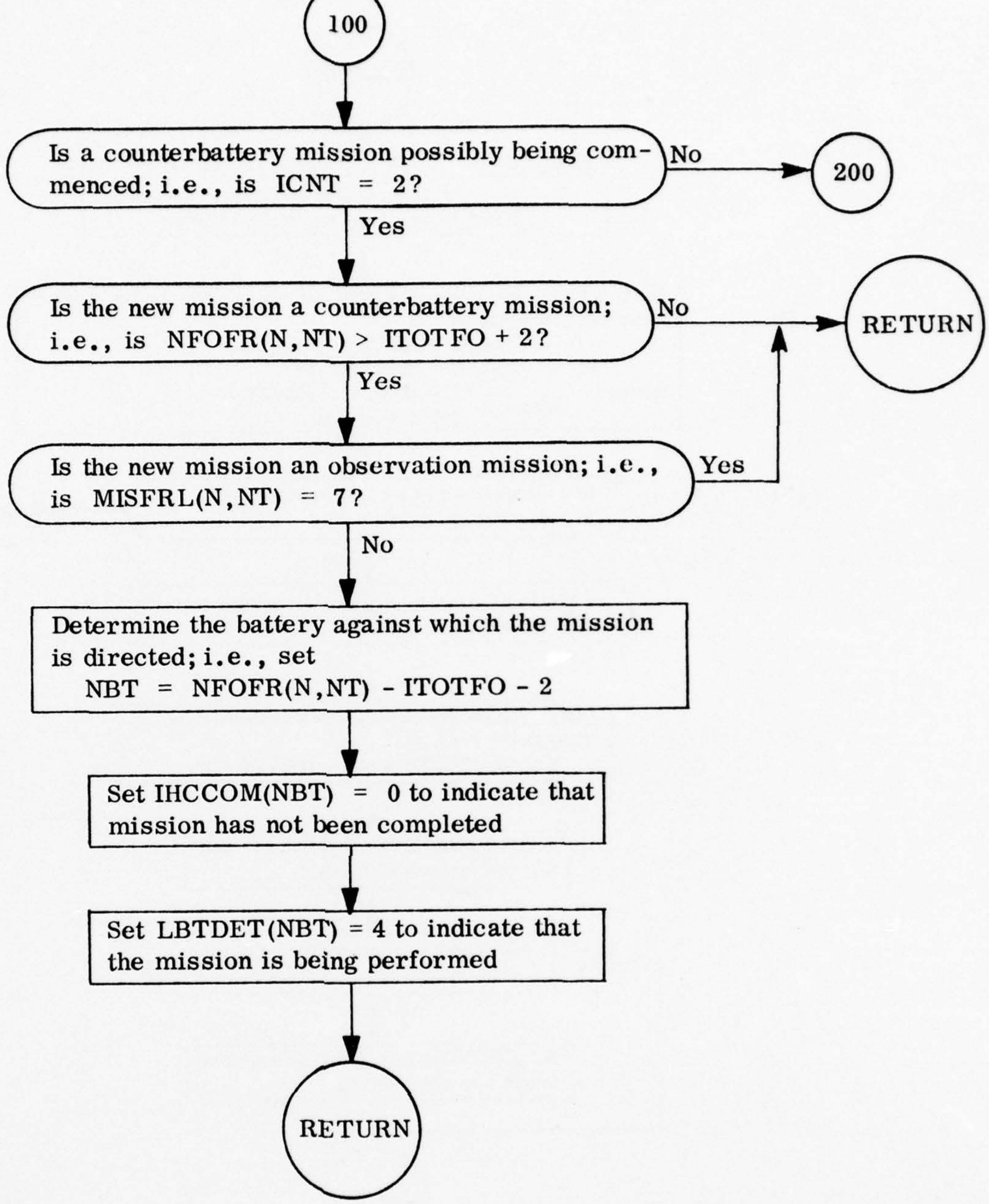
NUMBER	CBERR	NVOLM
NFOFR	MISFRL	YAIC
LWCOD	IUNACT	ECLOCK
YFB	XAIC	TMISUN
IHCCOM	KFO	XFB
LBTDET		

SUBROUTINES REQUIRED:

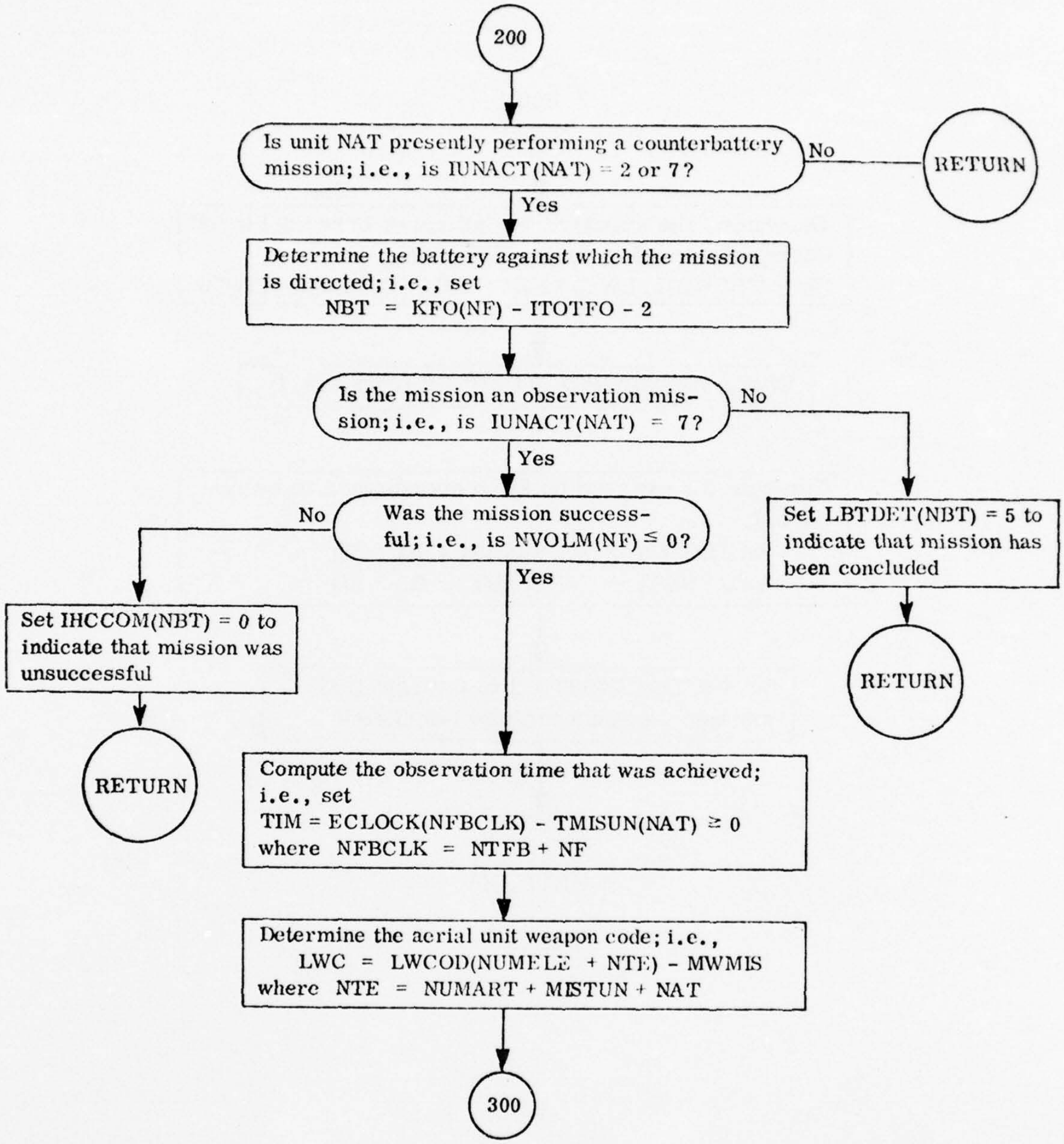
RANND



Subroutine CBCONT: Termination of Counterbattery Missions



Subroutine CBCONT: Continued



Subroutine CBCONT: Continued

300

Determine the standard deviations of error achieved during observation during TIM; i.e.,
SD = CBERR(1, LWC) * EXP(-TIM * CBERR(2, LWC))

Obtain two numbers R1 and R2 from N(0, 1)

Compute the errored position coordinates to be reported; i.e.,

XAIC(NBT) = XFB(NBT) + R1 * SD
YAIC(NBT) = YFB(NBT) + R2 * SD

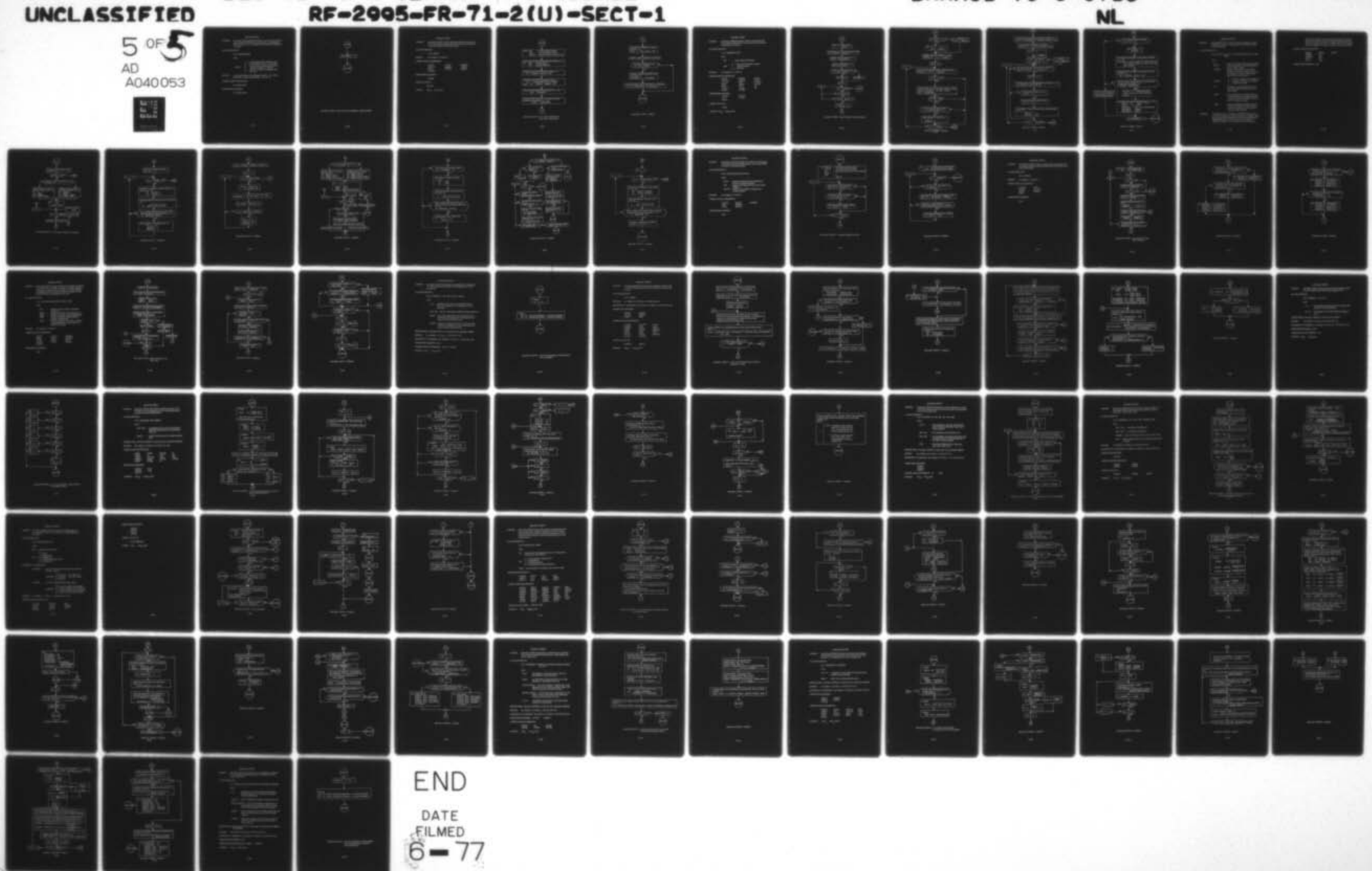
Set IHCCOM(NBT) = 2 to indicate that mission was successfully completed

RETURN

Subroutine CBCONT: Continued

AD-A040 053 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM). VOLUME 2, SECTION--ETC(U)
DEC 71 G M CLARK, R J WILHELM DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-2(U)-SECT-1 NL

5 OF 5
AD
A040053

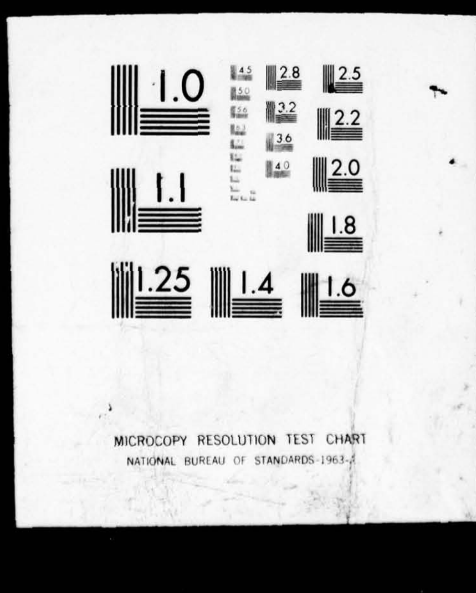


END
DATE
FILMED
6-77

SIFIED

5 OF 5

AD
A040053



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Subroutine CBFIR

PURPOSE: Subroutine CBFIR determines whether an aerial section can fire upon an assigned artillery unit target during a counterbattery attack mission. If so, parameters of the attack are defined and recorded.

CALLING SEQUENCE:

CALL CBFIR(ITGASN)

where

$$
\text{ITGASN} = \begin{cases} 0 & \text{section should seek a defensive position because no attack can take place} \\ 1 & \text{section cannot yet attack the artillery unit but no defense is required} \\ 2 & \text{section can attack the assigned artillery unit} \end{cases}
$$

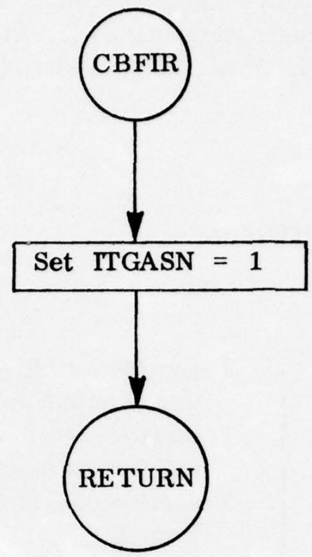
METHOD: This program has not been designed in detail. The calling parameter is always returned with a value of one.

COMMON AREAS REFERENCED:

To be determined

SUBROUTINES REQUIRED:

To be determined.



Subroutine CBFIR: Aerial Section Counterbattery Target Selection

Subroutine CBOBS

PURPOSE: Subroutine CBOBS is used to determine whether an aerial unit with a counterbattery observation mission has successfully observed the assigned artillery unit target.

CALLING SEQUENCE:

CALL CBOBS

METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

CBDET	LWCOD	NUMBER
ECLOCK	MANHEL	NVOLM
ETIM	NAVSEC	TMISUN
ICECOM		

SUBROUTINES REQUIRED:

RANND

CBOBS CALLED BY:

HELFIR

LENGTH: $2D8_{16} = 728_{10}$ bytes

CBOBS

Define ICE = current element number
ISEC = current element's section
MANUN = current element's maneuver unit

Determine the aerial section and unit numbers; i.e.,
set NSEC = NAVSEC(ISEC)
NAT = MANHEL(MANUN)

Determine the fire support firer number; i.e.,
NF = NUMART + ITOTLN + NAT

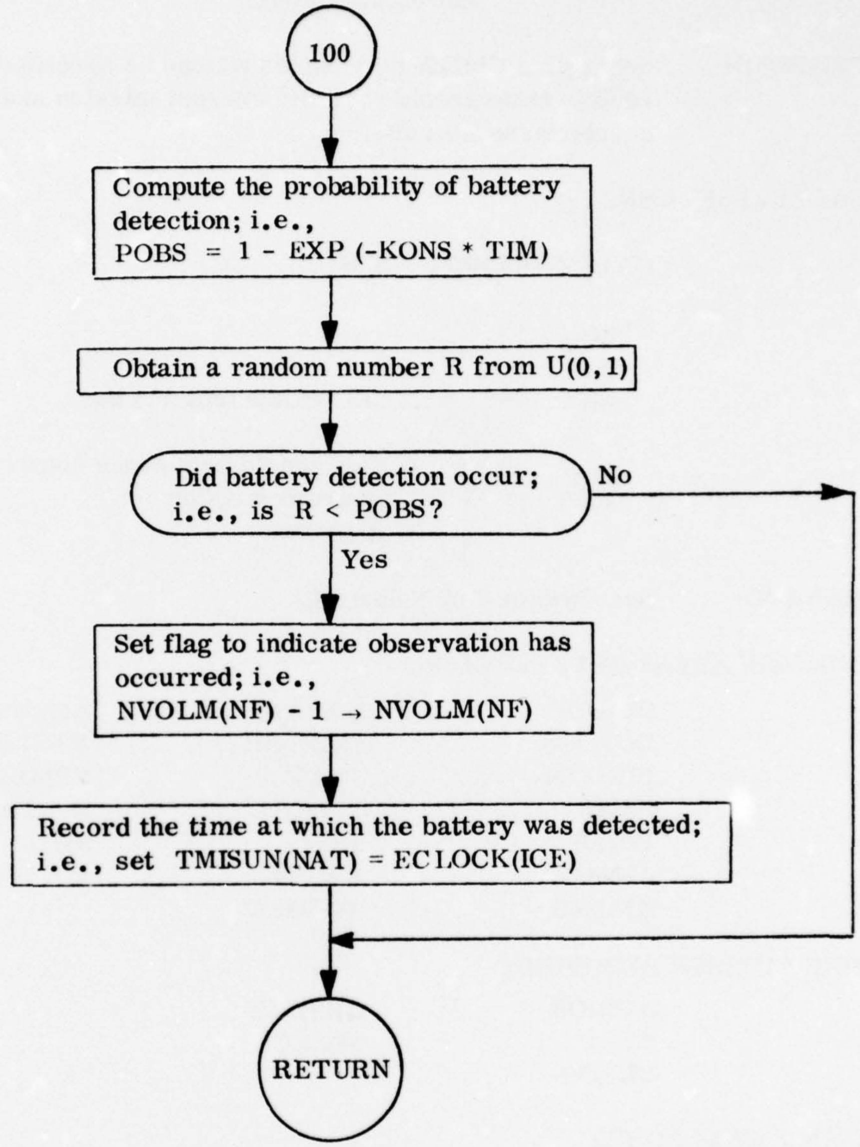
Determine the aerial unit weapon code; i.e.,
LWC = LWCOD(NUMELE+NT) - MWMIS
where NT = NUMART + MISTUN + NAT

Determine the length of the last event; i.e., set
TIM = ETIM(ICE)

Determine the constant for the battery detection probability distribution; i.e., set
KONS = CBDET(LWC)

100

Subroutine CBOBS: Aerial Unit Counterbattery Observation Performance



Subroutine CBOBS: Continued

Subroutine CMMIS

PURPOSE: Subroutine CMMIS determines whether a specified aerial vehicle team should abort its current mission and initiate a countermeasure mission.

CALLING SEQUENCE:

CALL CMMIS(NAT,ICM)

where

NAT = aerial vehicle unit number

ICM = { 1 NAT should commence countermeasure mission; 0 otherwise

METHOD: See Chapter 4 of Volume 1.

COMMON AREAS REFERENCED:

DURFB	KPRIOR	NOBVH
ECLOCK	MANLDR	NTELE
ICECOM	LDET	NUMBER
IMIST	LMANU	NVOLM
INART	LSEC	XD
JUNACT	LWCOD	YD
KMANU	NAVSEC	

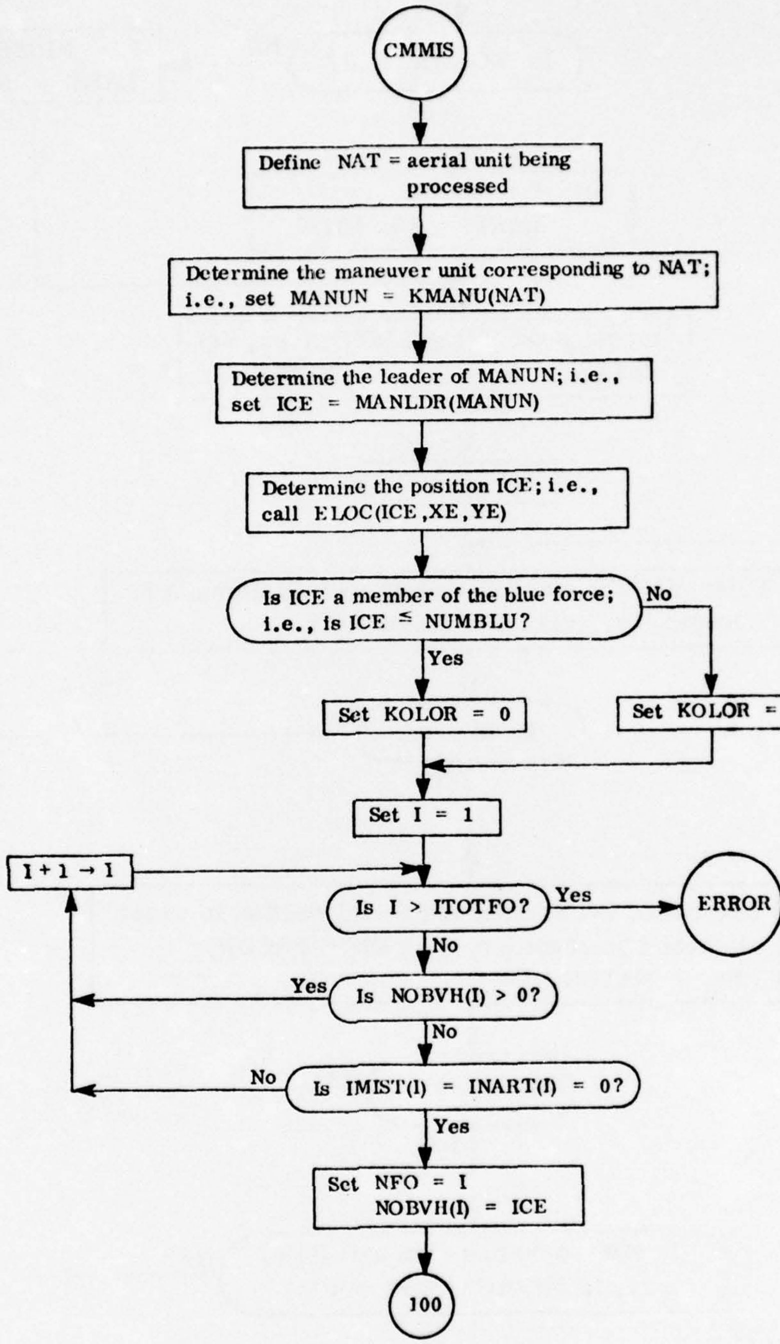
SUBROUTINES REQUIRED:

APRIOR	GETDET
ELOC	SELECA
ERROR	

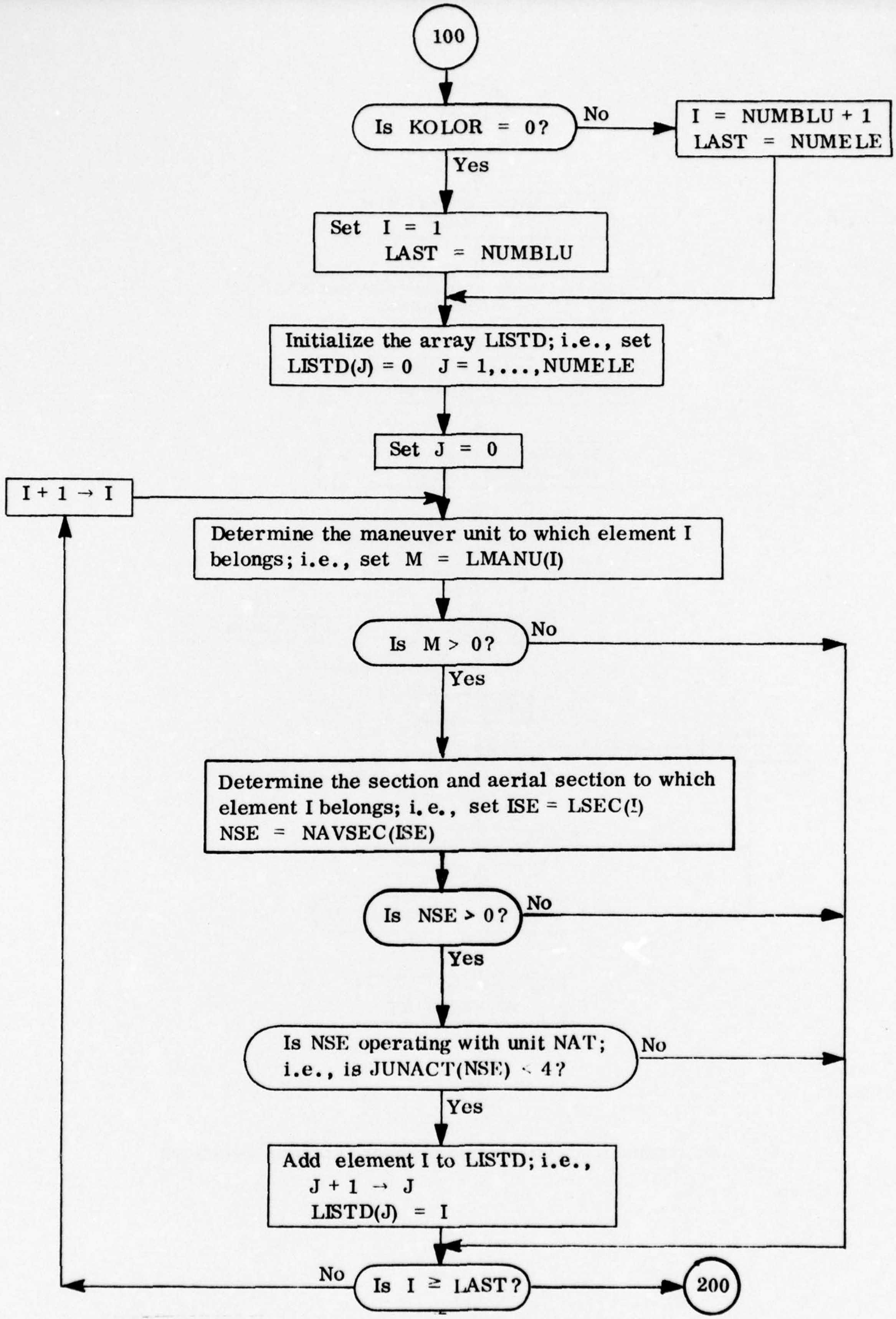
CMMIS CALLED BY:

ATDEC

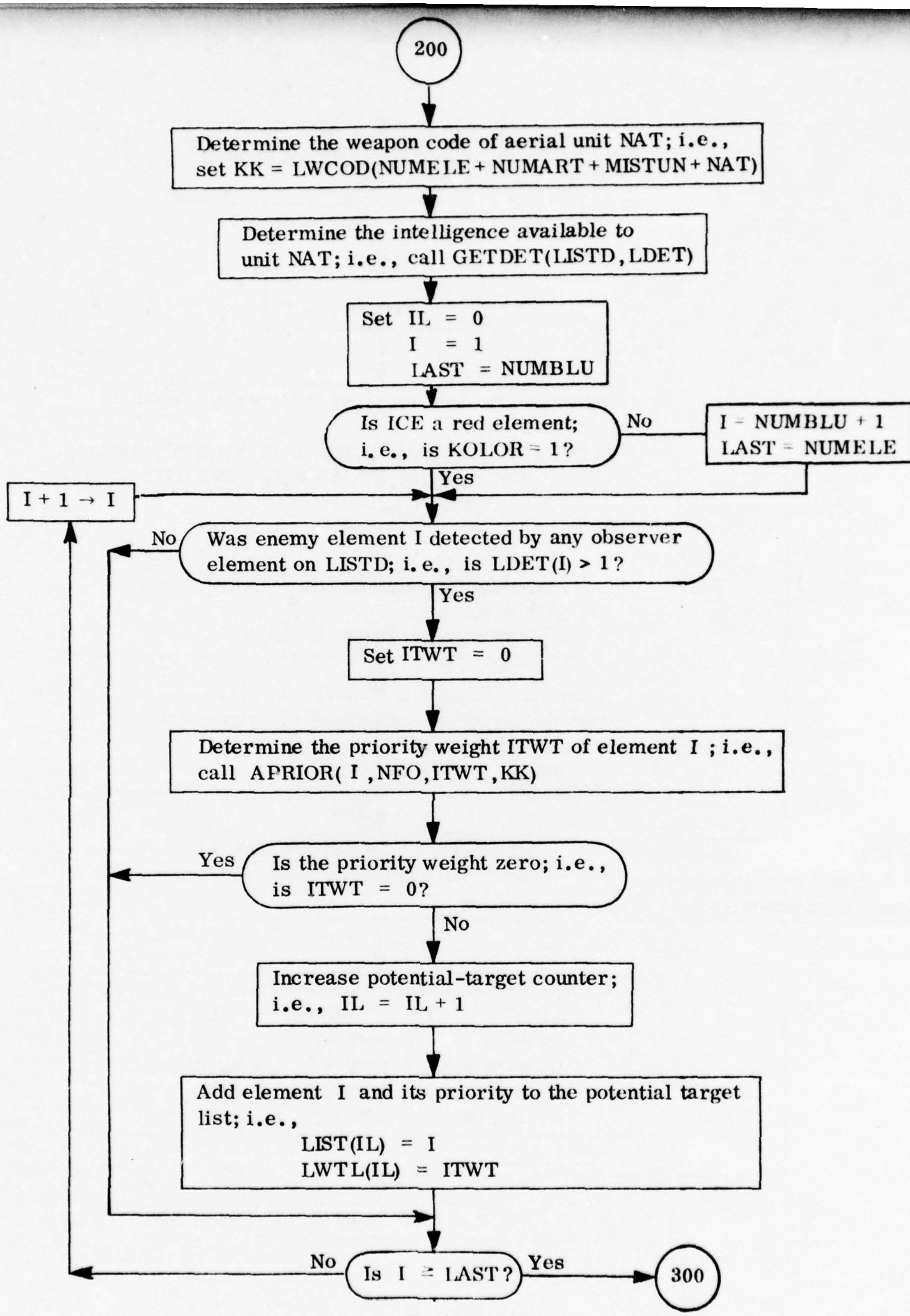
LENGTH: $968_{16} = 2408_{10}$ bytes



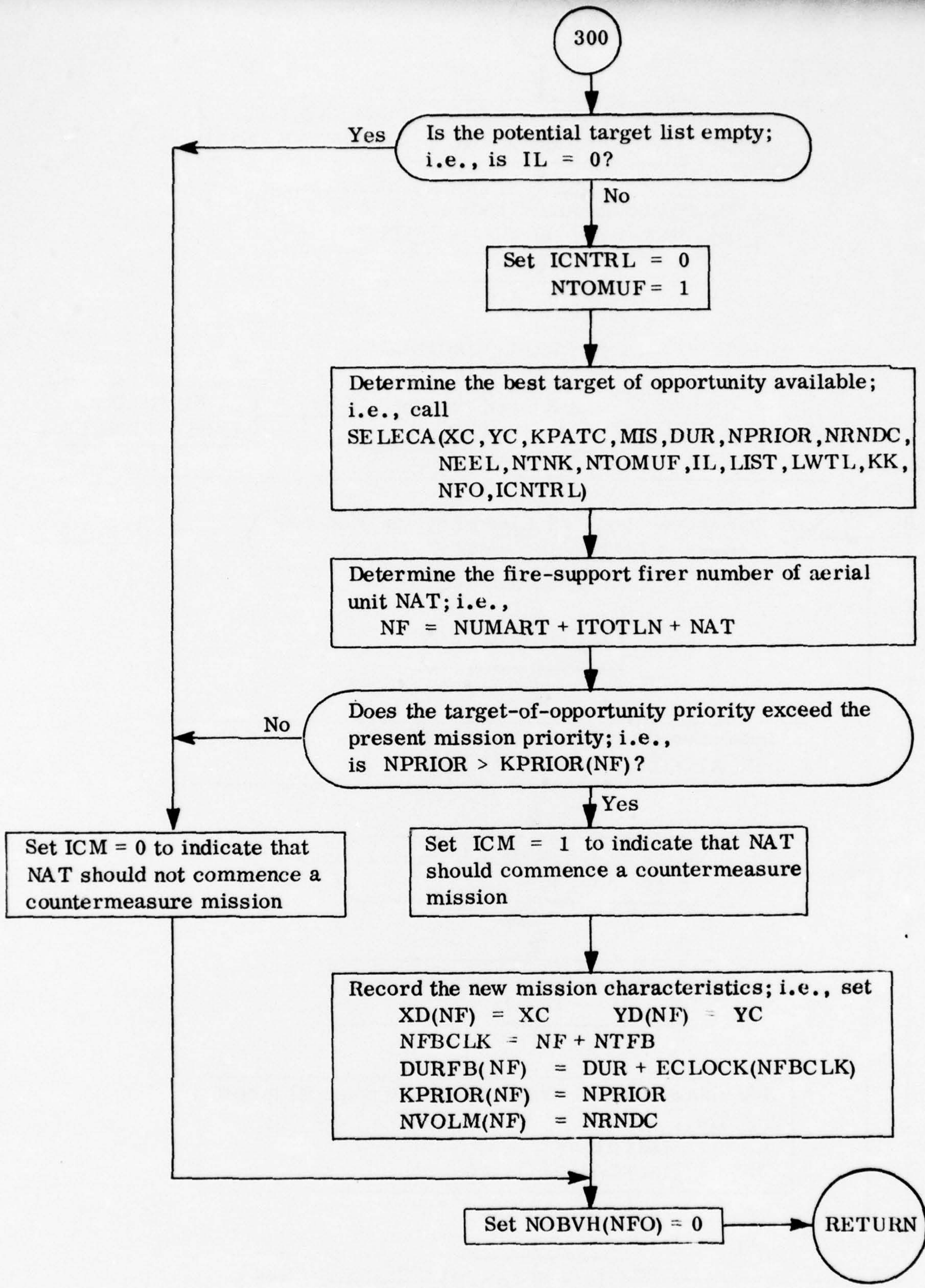
Subroutine CMMIS: Countermeasure Mission Decisions



Subroutine CMMIS: Continued



Subroutine CMMIS: Continued



Subroutine CMMIS: Continued

Subroutine CNFLCT

PURPOSE: Subroutine CNFLCT is called to determine whether a conflict exists between new fire requests and those already being processed or executed.

CALLING SEQUENCE:

CALL CNFLCT (KK, XC, YC, NCNTRL, KOLOR, IL1, LIST1, LIST2)

where

KK	=	weapon code of fire being requested (input)
XC, YC	=	position coordinates of target reported in the fire request (input)
NCNTRL	=	number of the unit whose current fire requests and missions in progress are to be examined. If zero, all fire-support units will be examined (input)
KOLOR	=	0 if the fire request was initiated by a member of the blue force (input) 1 if otherwise (input)
IL1	=	the number of current requests and missions that conflict with the new fire request (output)
LIST1	=	array containing unit numbers with conflicting requests or missions in progress. A negative valued entry indicates the mission is in progress (output)
LIST2	=	array containing the positions of the conflicting fire requests on the unit's fire-request lists (nonzero only for positive entries in LIST2) (output).

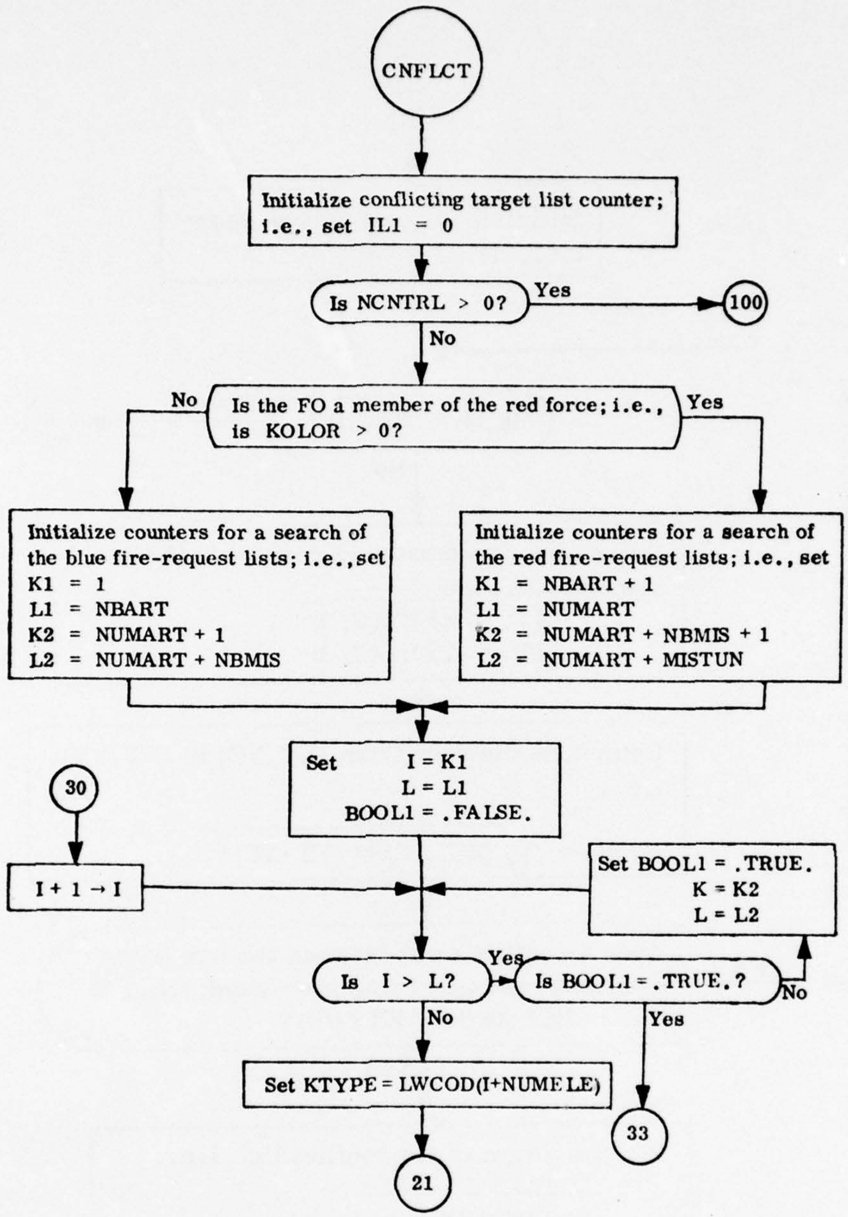
METHOD: If NCNTRL is zero, all artillery and MISTIC fire direction center fire-request lists will be examined for conflicts. Then the missions being executed by all artillery firing batteries, MISTIC launchers and aerial teams will be examined. If NCNTRL is positive, the fire-request list for artillery or

MISTIC unit NCNTRL will be examined, followed by an examination of the mission or missions currently being executed by the firer or firers of the unit. A conflict exists if the requested target coordinates are in close proximity to an existing target.

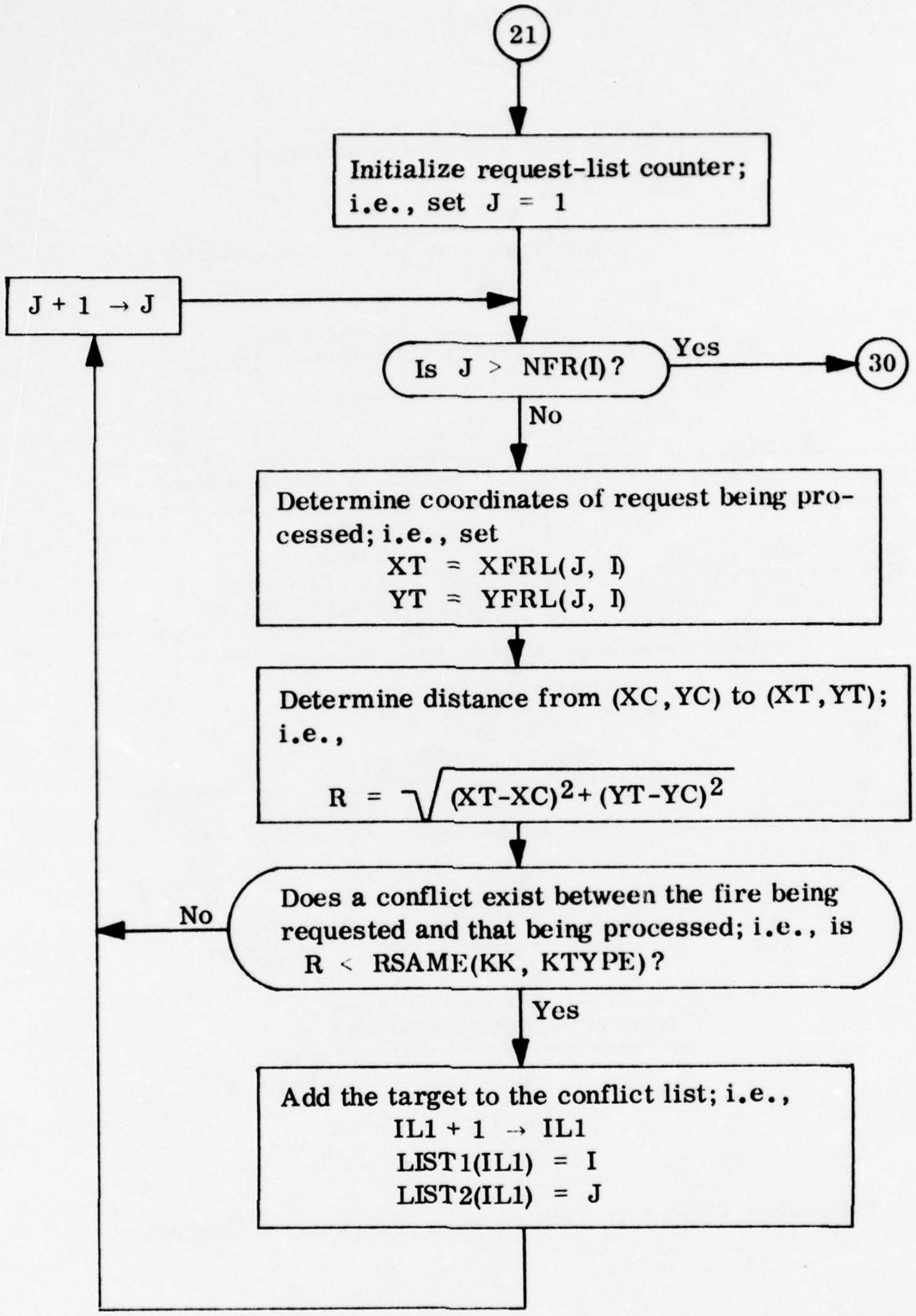
COMMON AREAS REFERENCED:

IFBMIS	RSAME	MISTYP
NMISUN	XD	
NUMBER	XFRL	
LWCOD	YD	
NFR	YFRL	

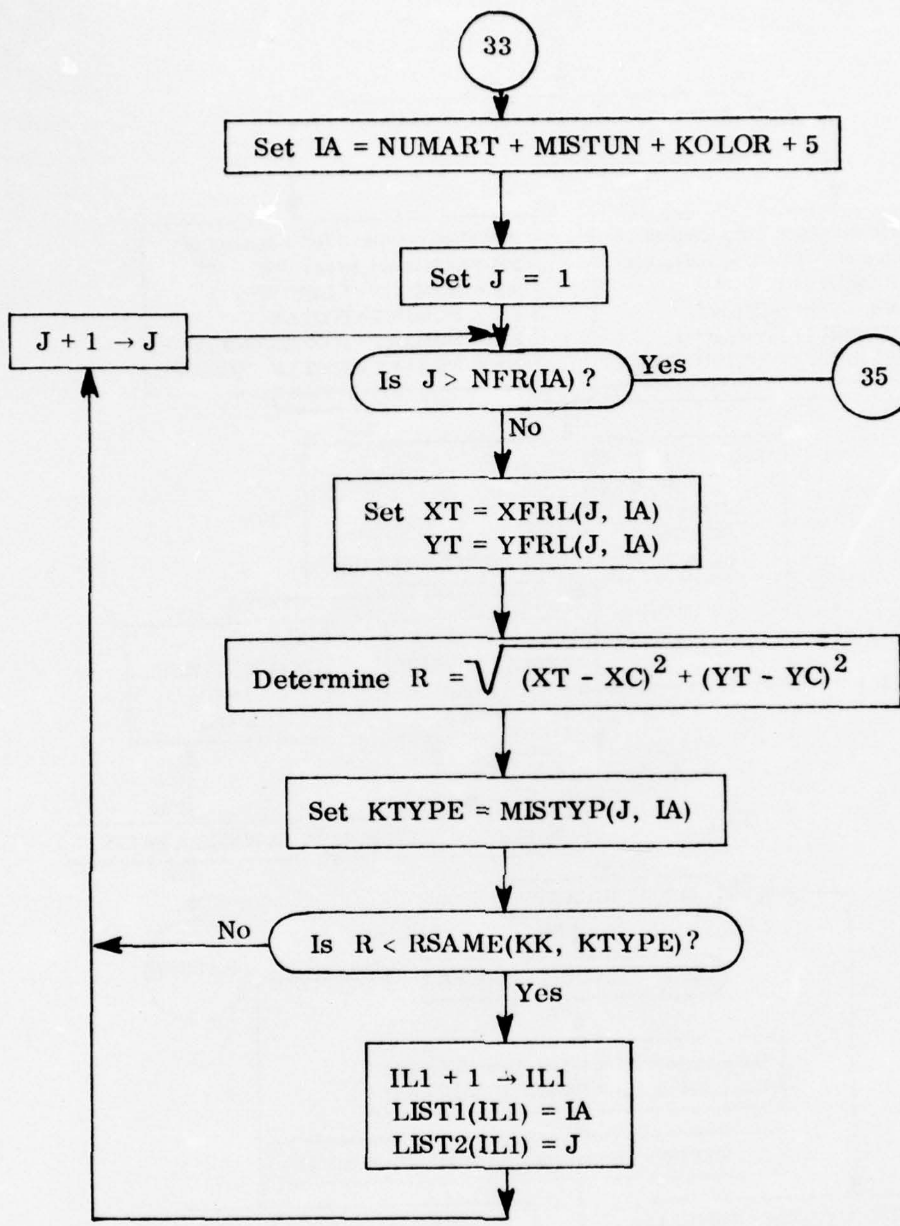
SUBROUTINES REQUIRED: None



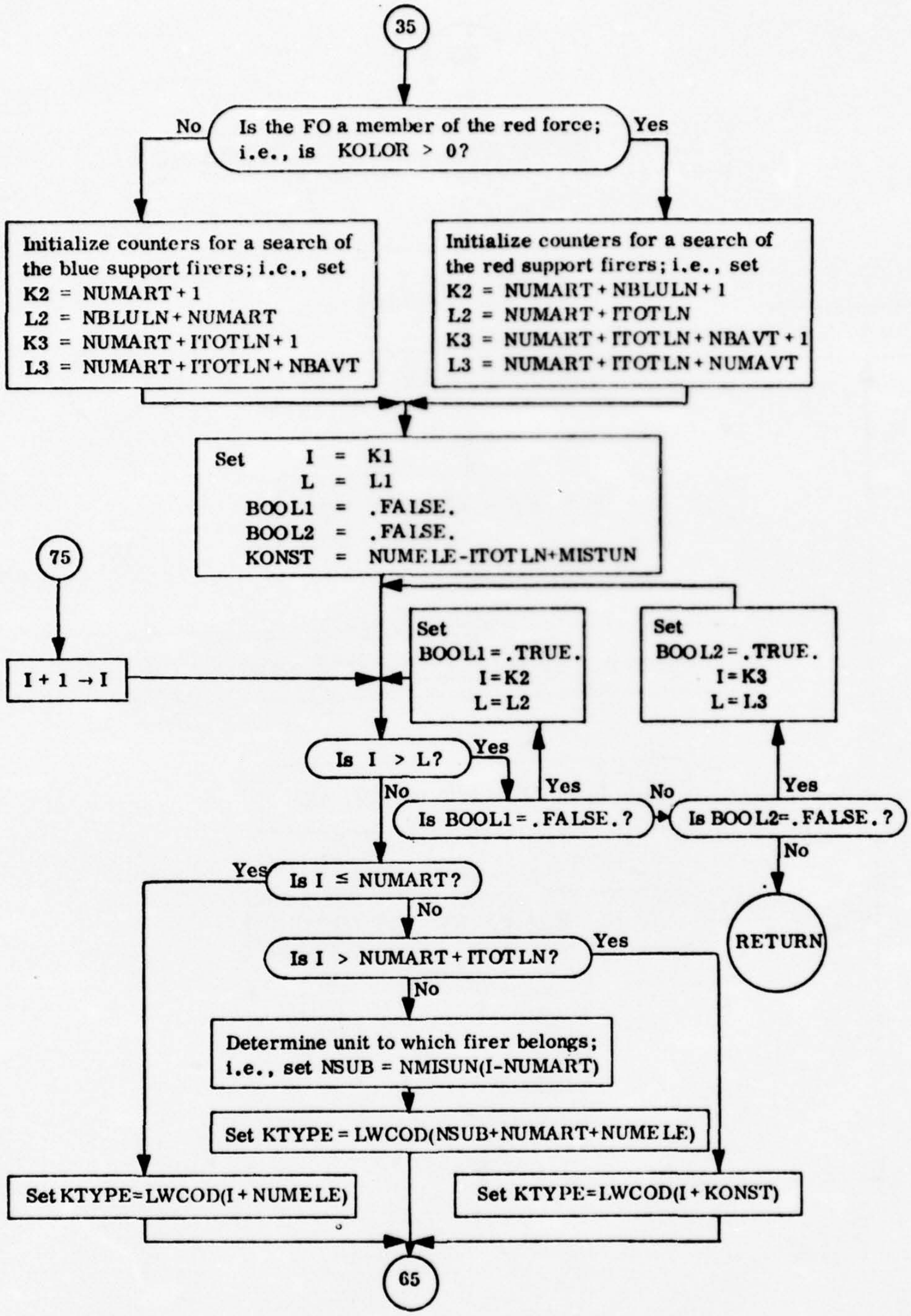
Subroutine CNFLCT: Determining Existence of Conflicts



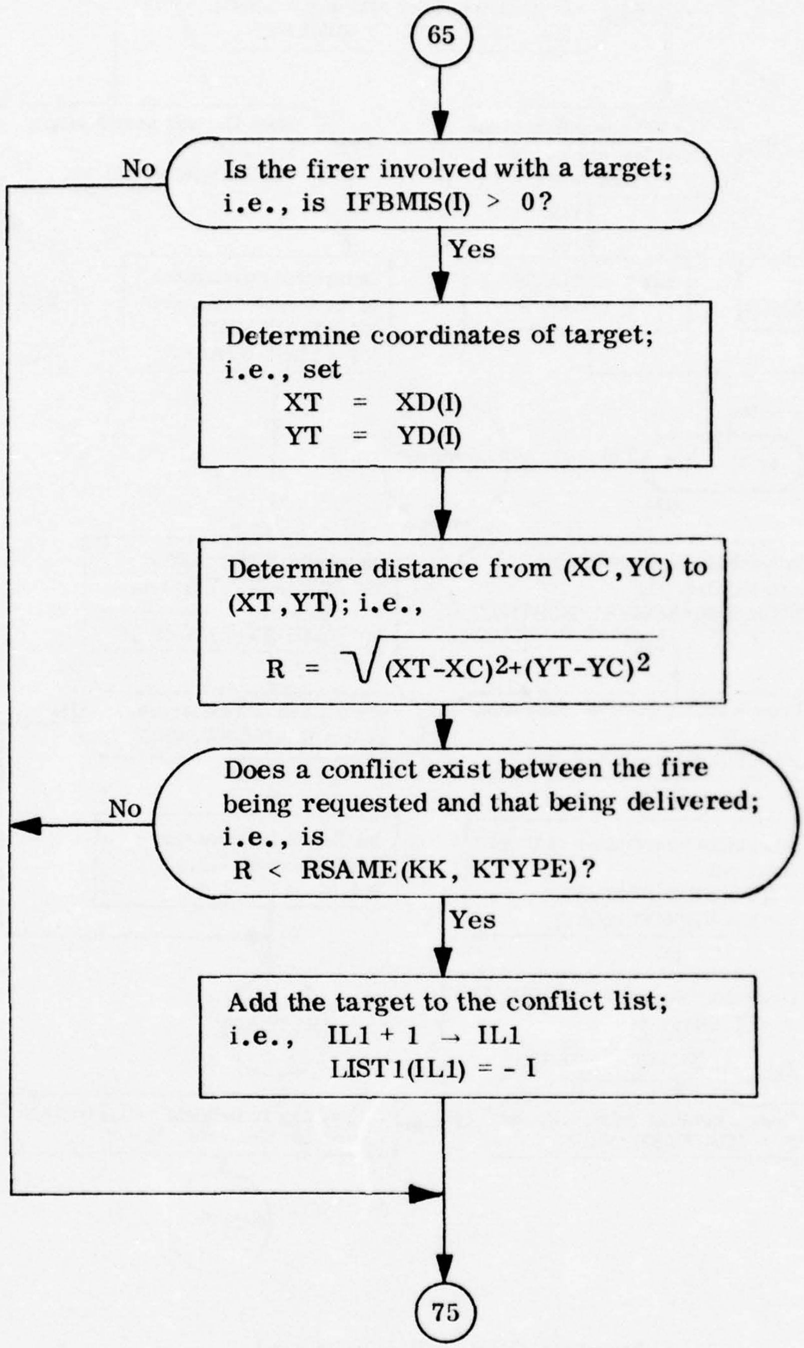
Subroutine CNFLCT: Continued



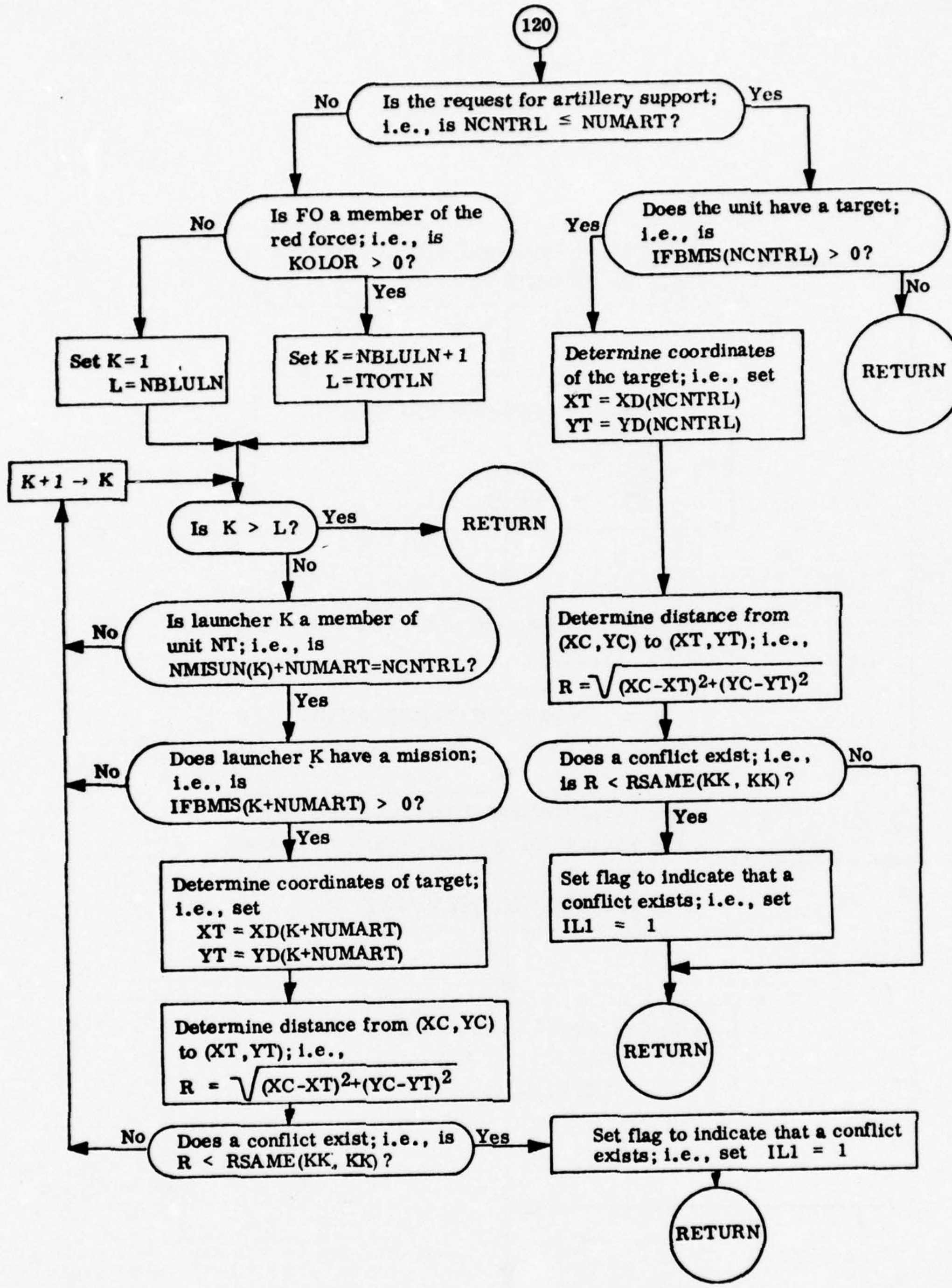
Subroutine CNFLCT: Continued



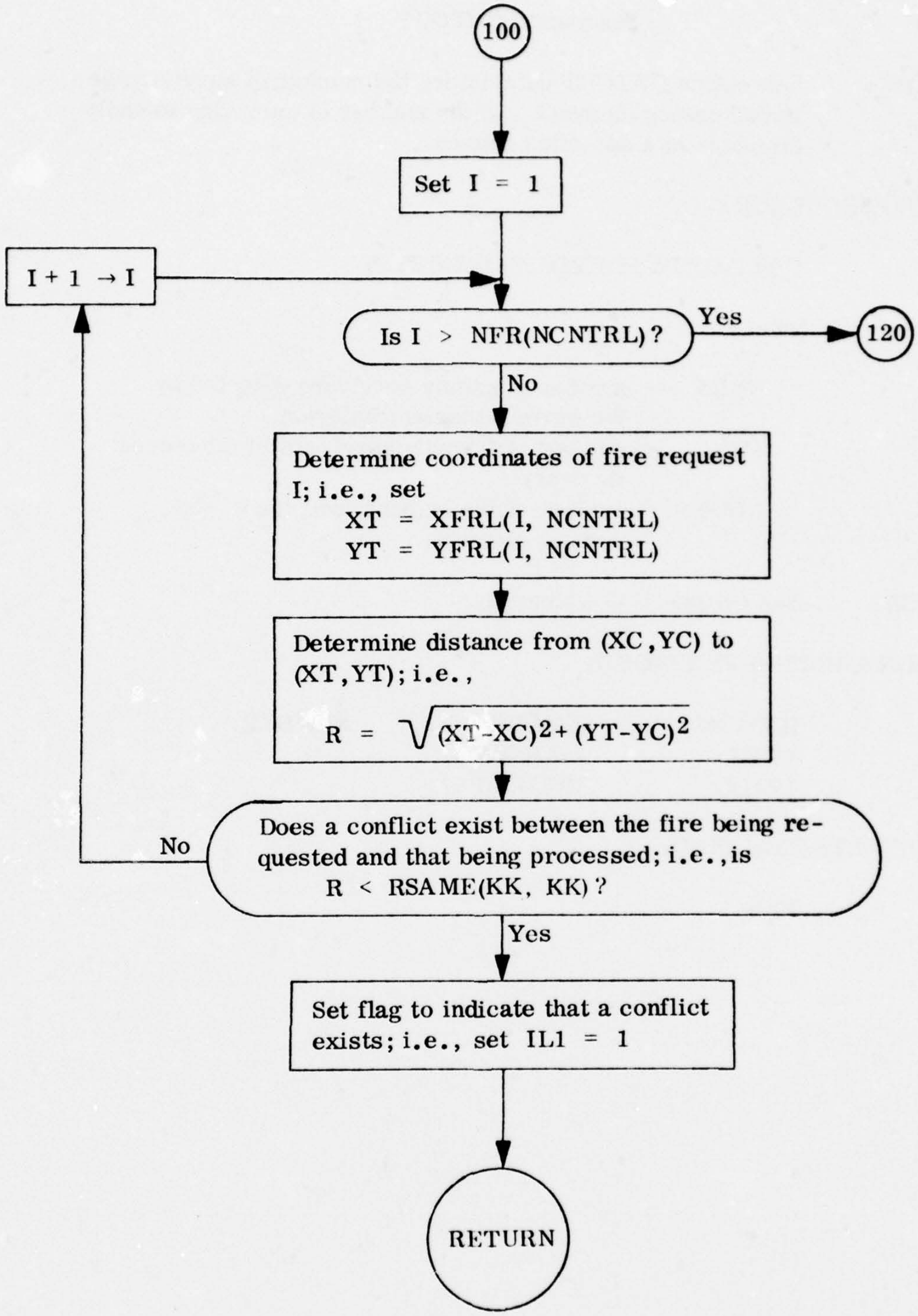
Subroutine CNFLCT: Continued



Subroutine CNFLCT: Continued



Subroutine CNFLCT: Continued



Subroutine CNFLCT: Continued

Subroutine CNTOUT

PURPOSE: Subroutine CNTOUT determines the number of surviving detected enemy elements and the number of surviving friendly elements at a specified outpost.

CALLING SEQUENCE:

```
CALL CNTOUT(KEDS,KFS,NOUTP)
```

where

KEDS	=	number of enemy survivors detected by the current element (integer)
KFS	=	number of friendly survivors at the outpost (integer)
NOUTP	=	number of the outpost occupied by the current element.

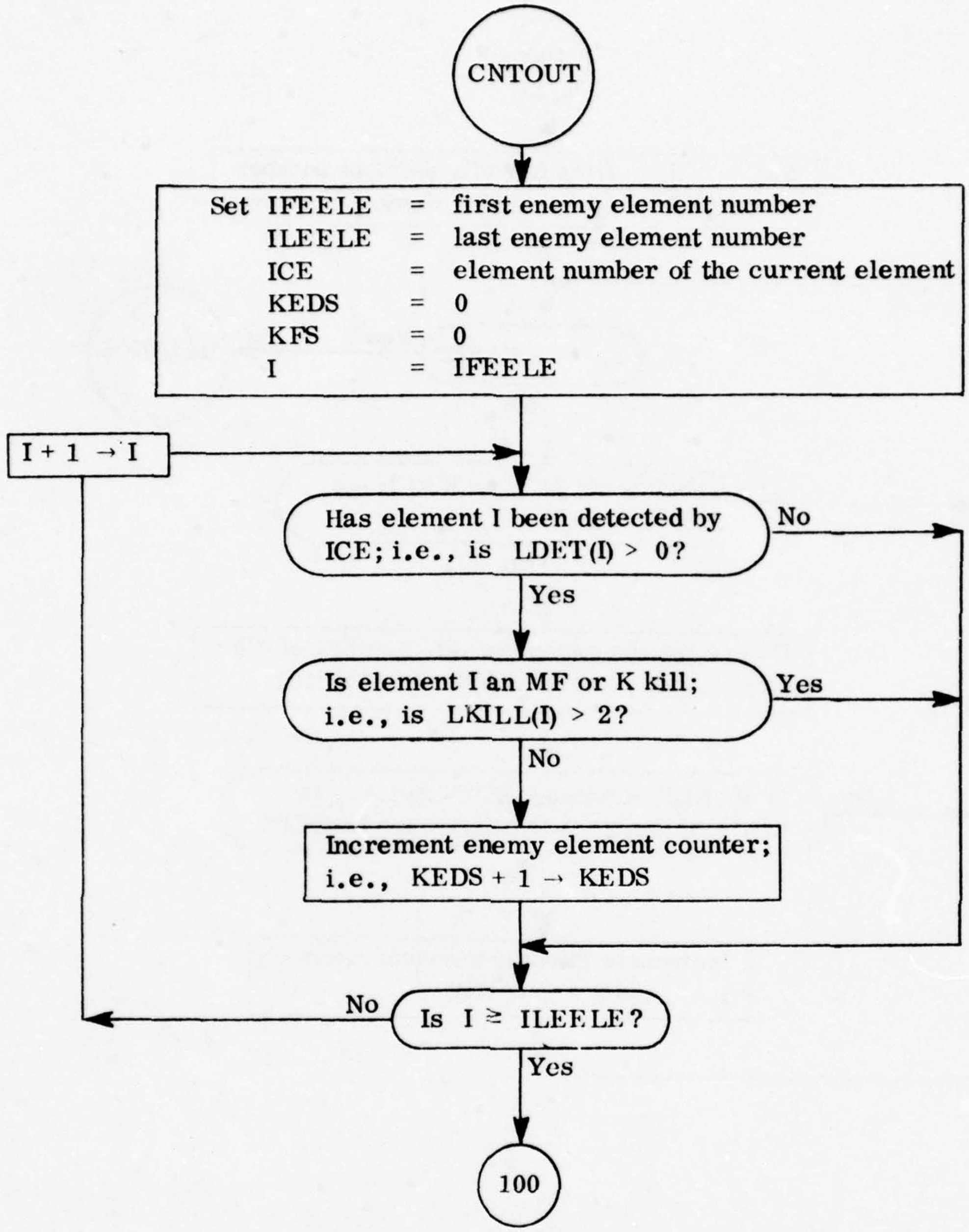
METHOD: See Chapter 9 of Volume 1.

COMMON AREAS REFERENCED:

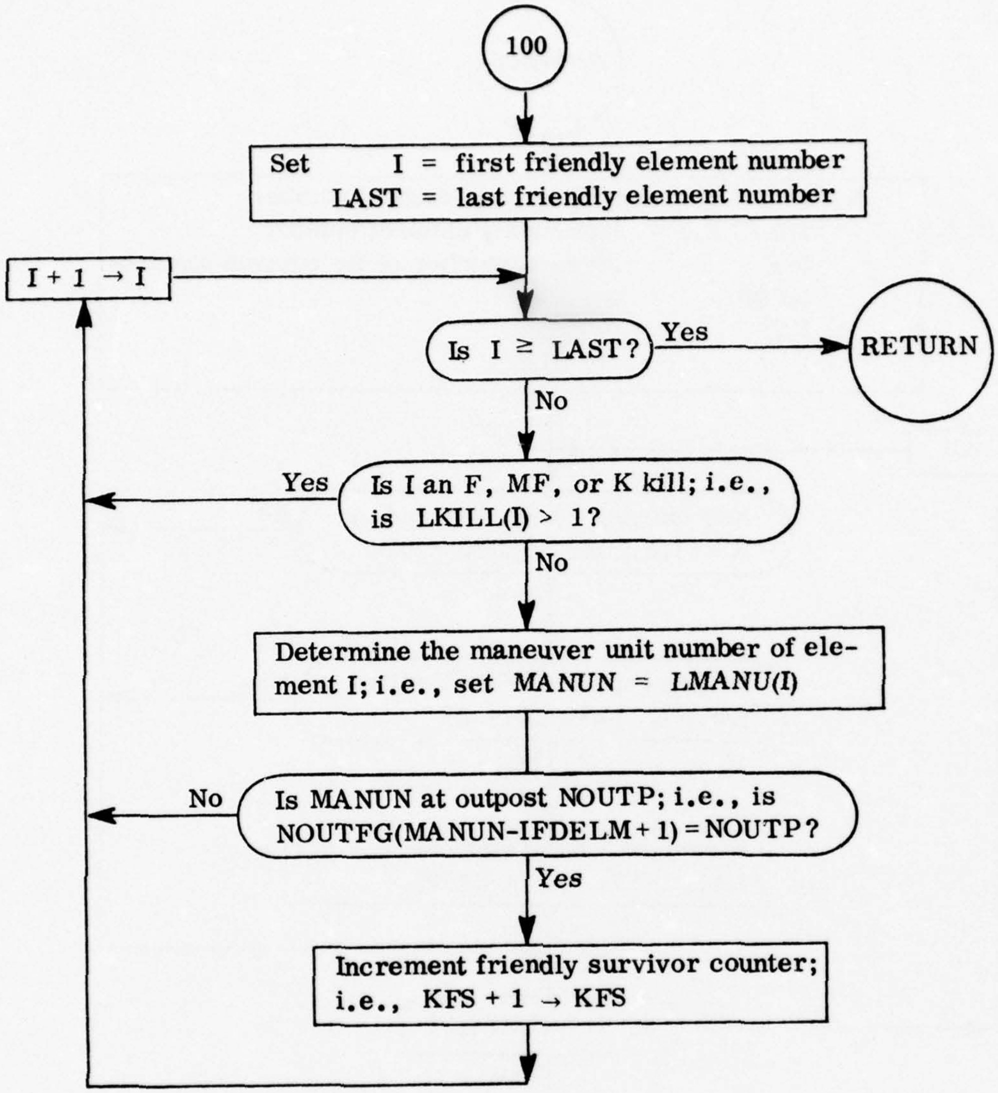
ICECOM	LMANU	NUMBER
LDET	MANEUV	
LKILL	NOUTFG	

SUBROUTINES REQUIRED:

None



Subroutine CNTOUT: Computing Outpost Survivors



Subroutine CNTOUT: Continued

Subroutine CONVRT

PURPOSE: Subroutine CONVRT is used to transform data coming from the aerial target selection model into data that is more usable in the aerial firing model.

CALLING SEQUENCE:

CALL CONVRT

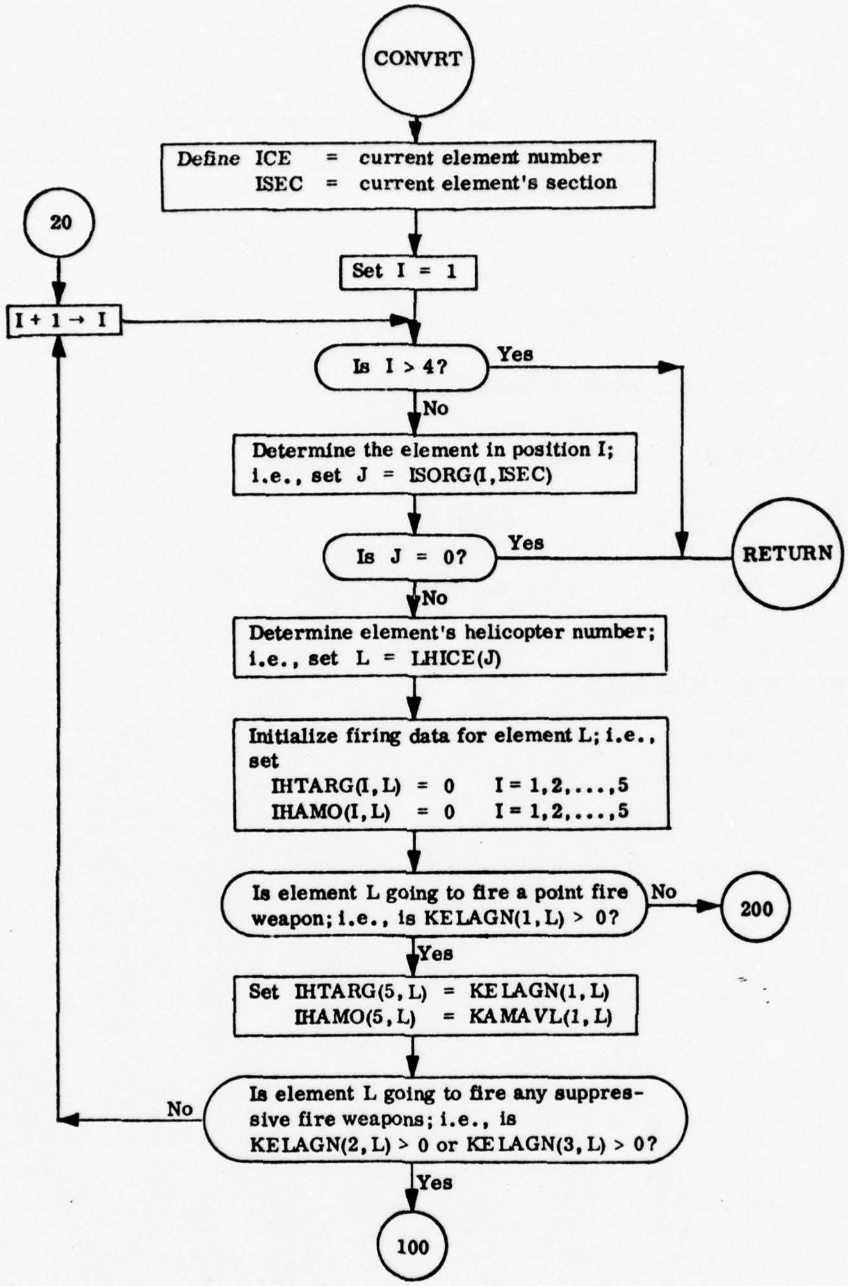
METHOD: See Chapter 6 of Volume 1.

COMMON AREAS REFERENCED:

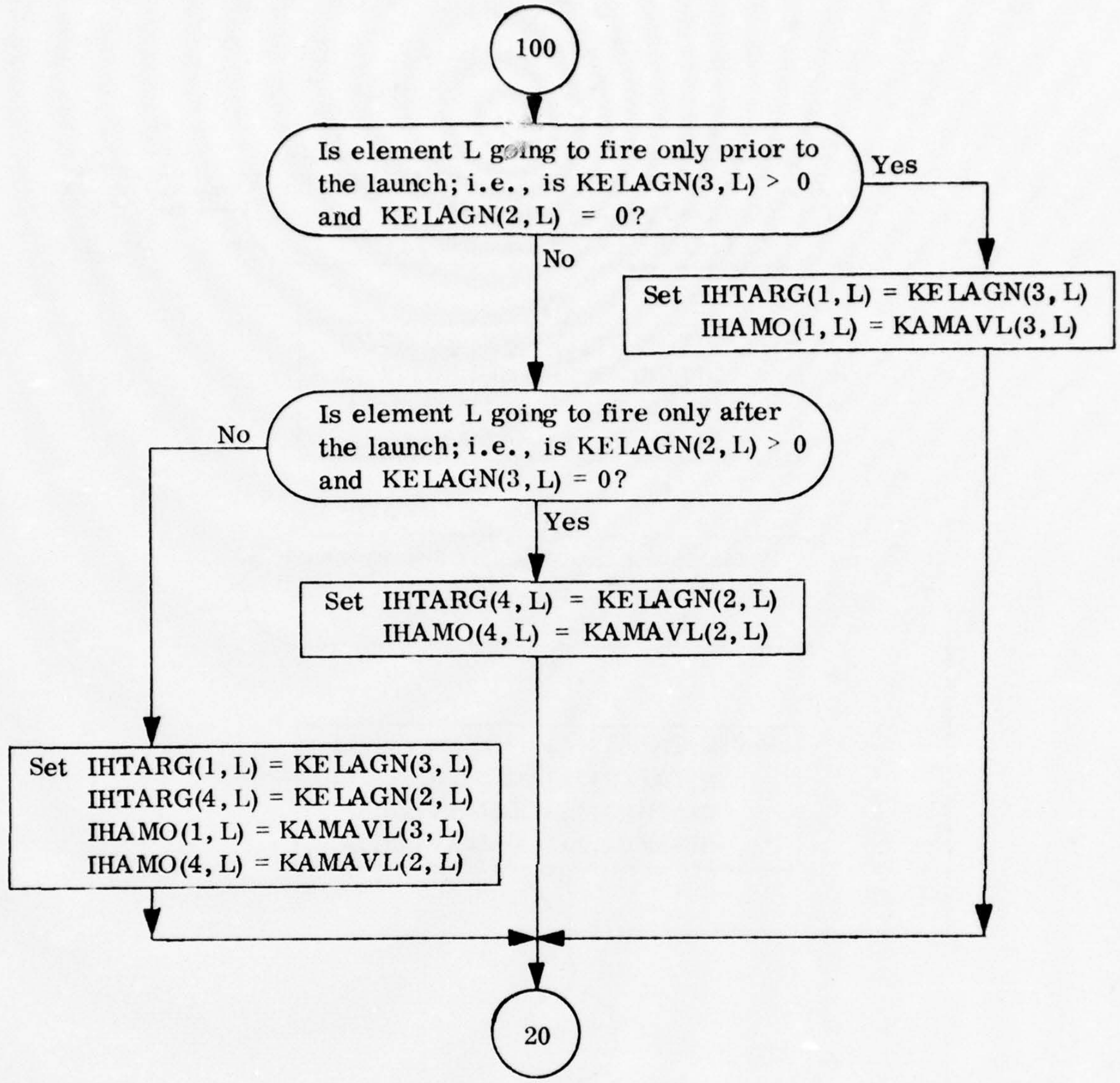
ICECOM	LHICE
IHTARG	KELAGN
IHAMO	KAMAVL
ISORG	

SUBROUTINES REQUIRED:

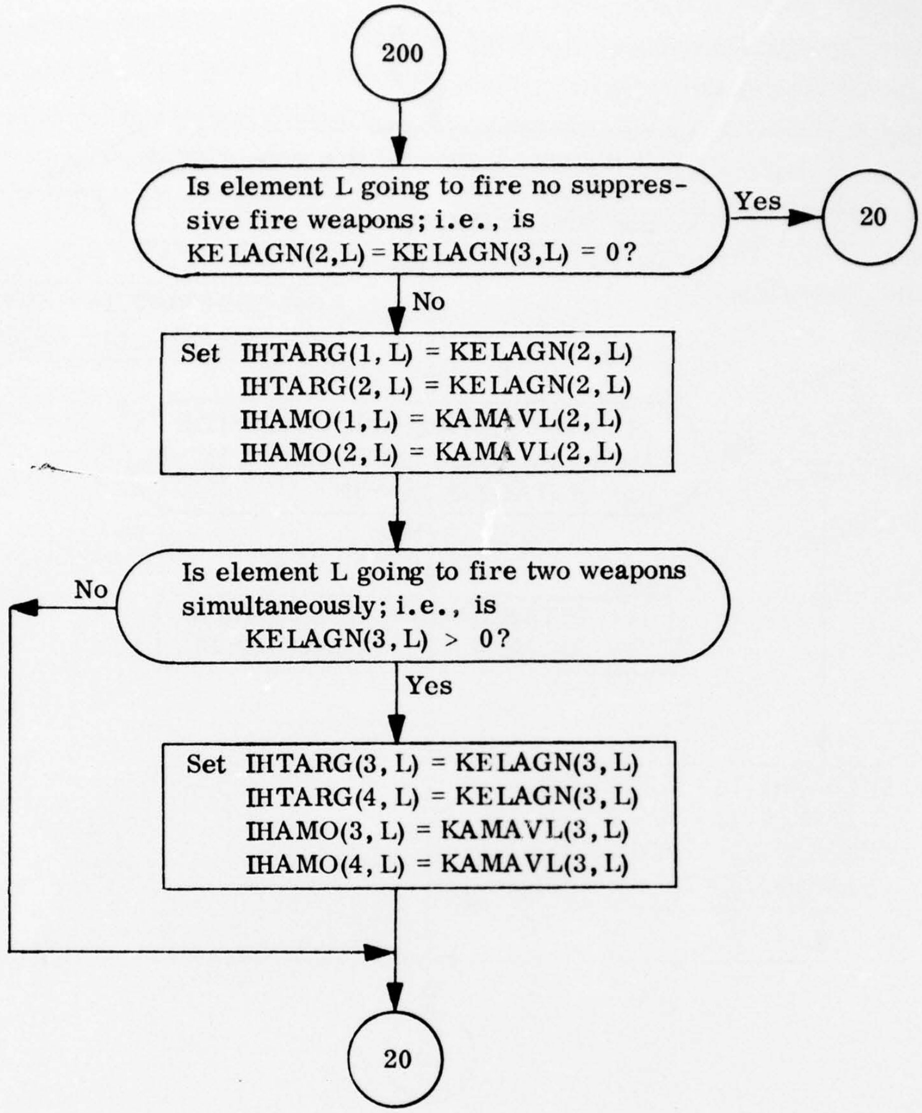
None



Subroutine CONVRT: Aerial Section Firing Data Conversion



Subroutine CONVRT: Continued



Subroutine CONVRT: Continued

Subroutine COUNT

PURPOSE: Subroutine COUNT is used to determine the number of sections in an aerial unit, the number of sections currently engaged in some type of firing activity, the number of sections retiring independently of the unit and the number of sections seeking defensive positions independently of the unit.

CALLING SEQUENCE:

```
CALL COUNT(NAT,IMEM,NRET,NDEF,NFIR)
```

where

NAT	=	number of aerial maneuver unit being processed
IMEM	=	number of sections in the maneuver unit
NRET	=	number of sections retiring independently
NDEF	=	number of sections seeking defensive positions independently
NFIR	=	number of sections with direct or indirect firing assignments or with requests for indirect fire support

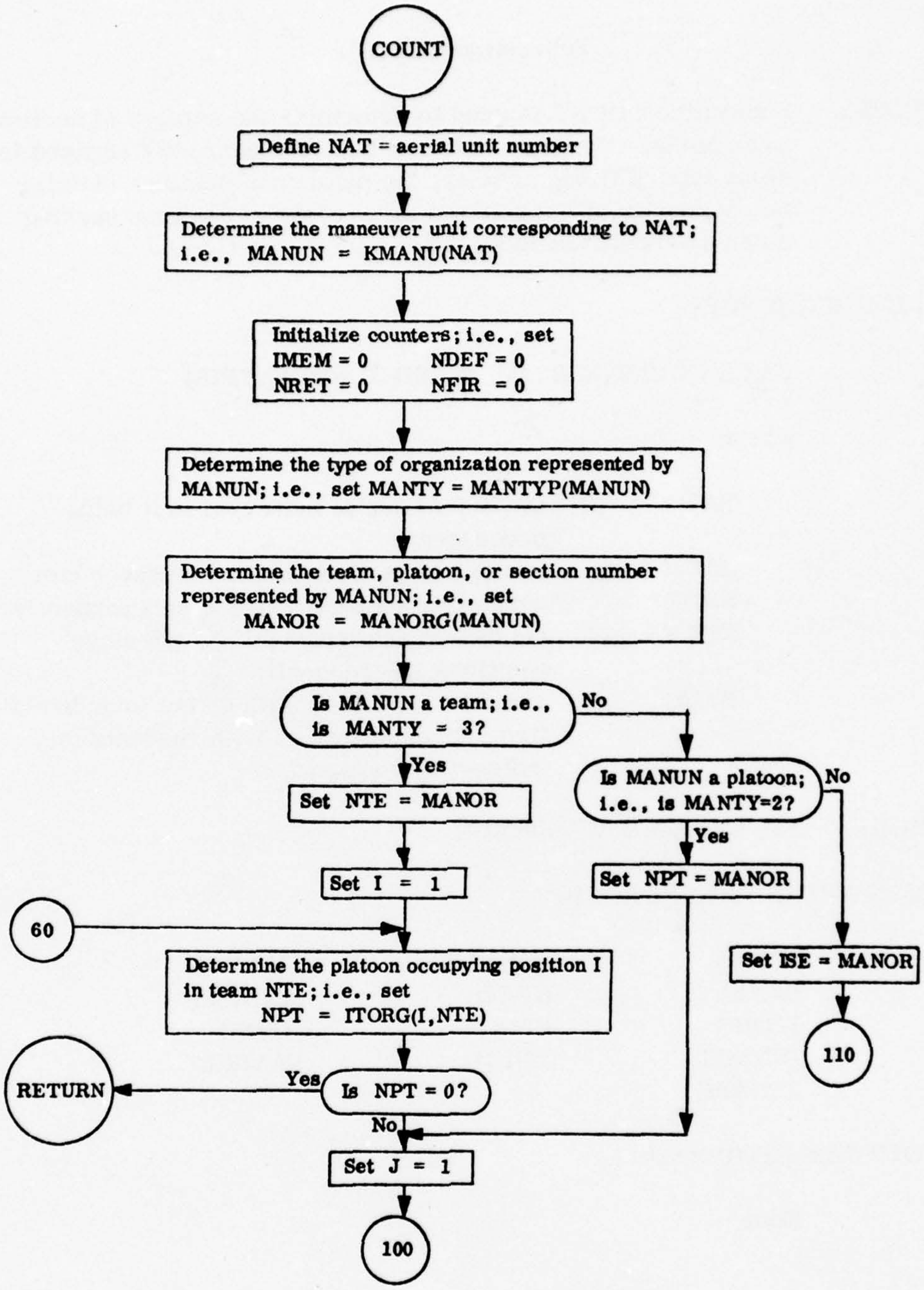
METHOD: See Chapter 4 of Volume 1.

COMMON AREAS REFERENCED:

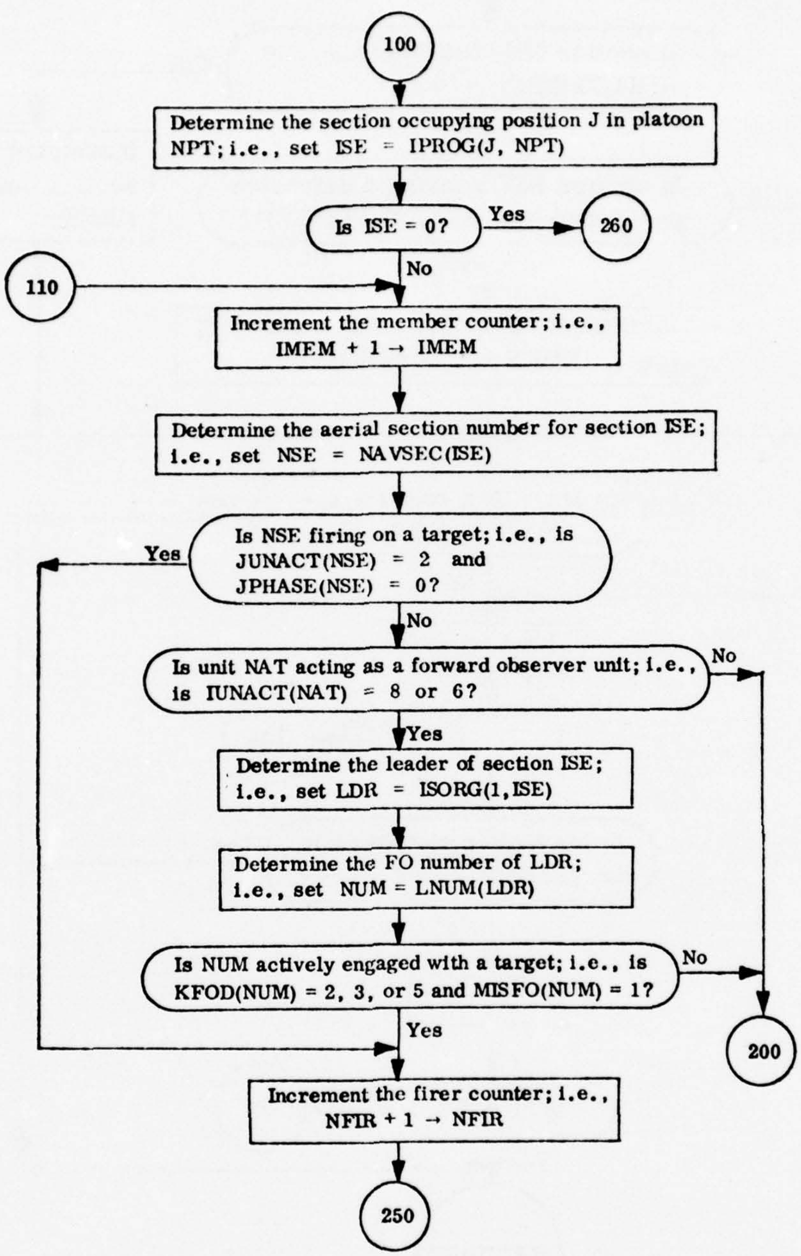
IPORG	JUNACT	MANORG
ISORG	KFOD	MANTYP
ITORG	KMANU	MISFO
IUNACT	LNUM	NAVSEC
JPHASE		

SUBROUTINES REQUIRED:

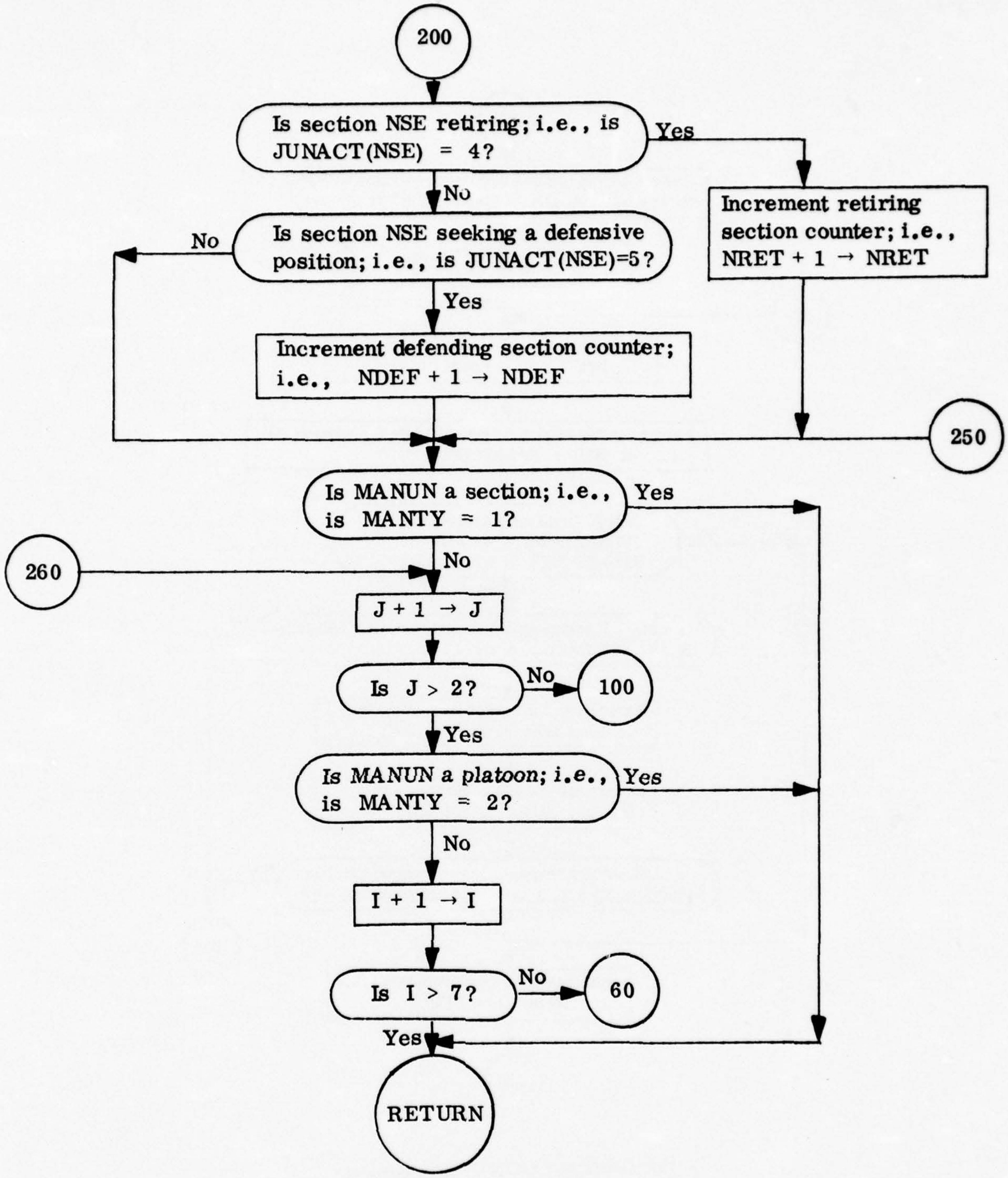
None



Subroutine COUNT: Status of Sections in an Aerial Unit



Subroutine COUNT: Continued



Subroutine COUNT: Continued

Subroutine CRTLOC

PURPOSE: Subroutine CRTLOC determines the battlefield X-Y coordinates of a route selection grid point for a dismounted crew-served weapon unit.

CALLING SEQUENCE:

CALL CRTLOC(I, J, XIJ, YIJ, SINAR, COSAR)

where

(I, J) = designator of the route selection grid point for which the battlefield X-Y coordinates are to be determined

(XIJ, YIJ) = the X-Y battlefield coordinates of grid point (I, J).

SINAR = sine of the angle between the current element's battlefield coordinates and the coordinates of the primary desired firing position.

COSAR = cosine of the angle between the current element's battlefield coordinates and the coordinates of the primary desired firing position.

RESTRICTIONS: Subroutine CRTLOC is called only from subroutine CSRTSL.

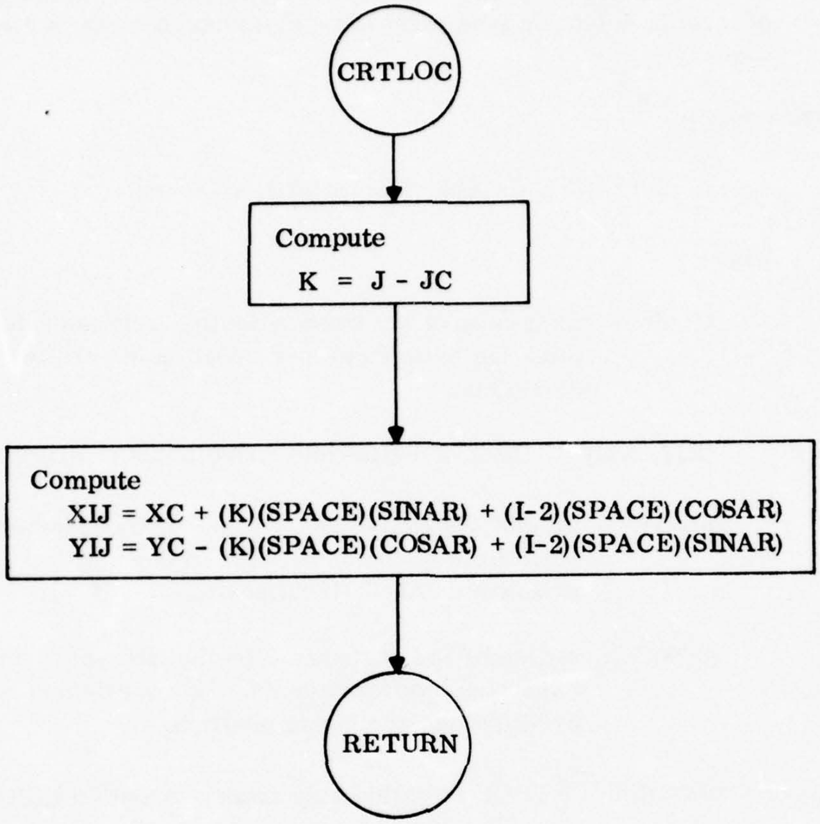
METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED: None

COMMON AREAS REFERENCED: CSWDF1, CSWIND

STORAGE: $25A_{16} = 602_{10}$ bytes



Subroutine CRTLOC: Crew-Served Weapons--Route Selection Grid Coordinates

Subroutine CSMOVE

PURPOSE: Subroutine CSMOVE determines the coordinates of the location to which a dismounted crew-served weapon unit moves during an event.

CALLING SEQUENCE:

CALL CSMOVE

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED:

CSTIME	PDPGET
ELVATE	PDPSET
ERROR	RGXY

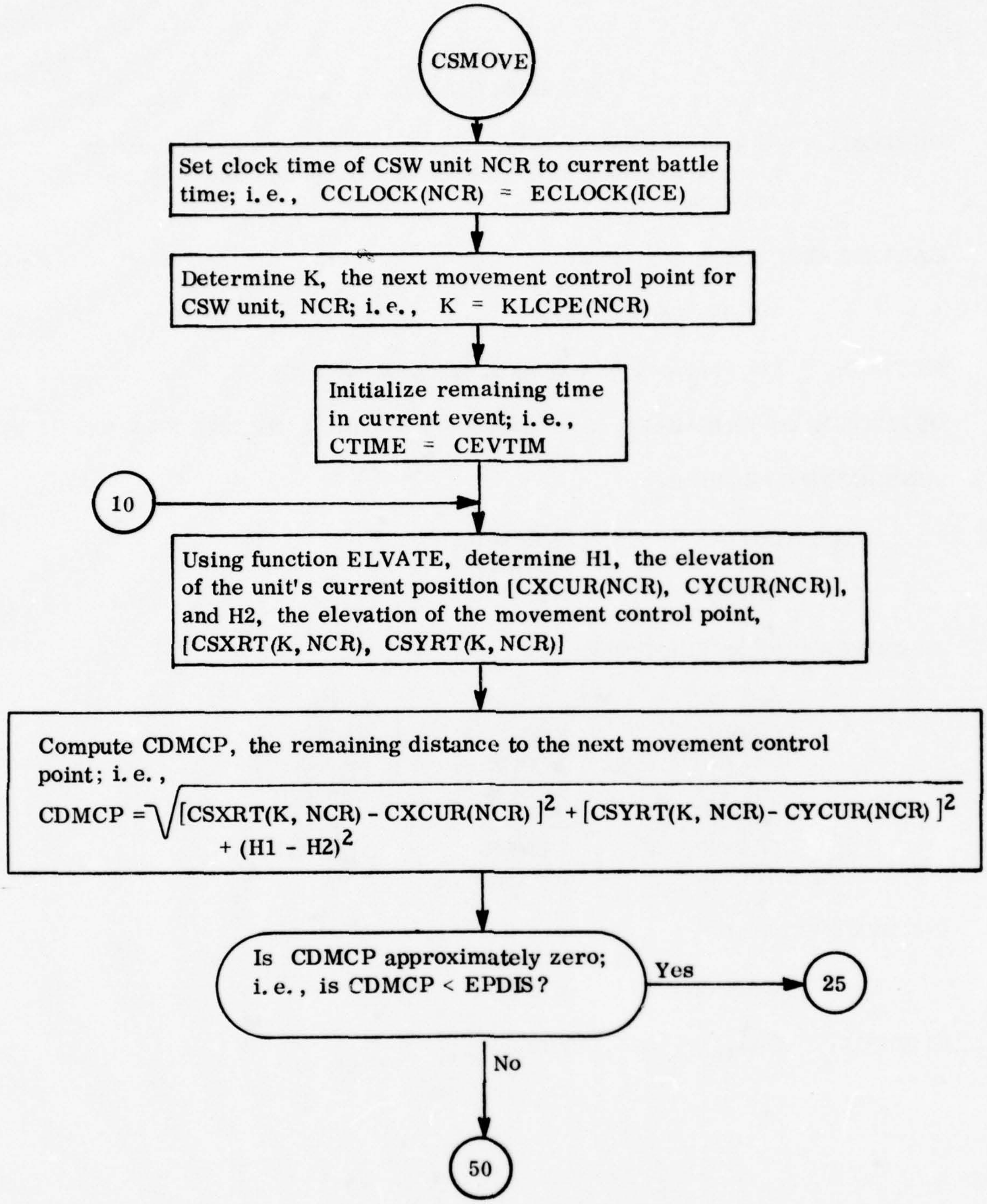
COMMON AREAS REFERENCED:

CCLOCK	ECLOCK	KLCPE
CSWDF1	EDIR	KSALT
CSWIND	ELOCX	KSMAKE
CSXRT	ELOCY	KSWHCH
CSYRT	ESPD	LDPC
CXCUR	EVBAR	MOVPAR
CYCUR	ICECOM	

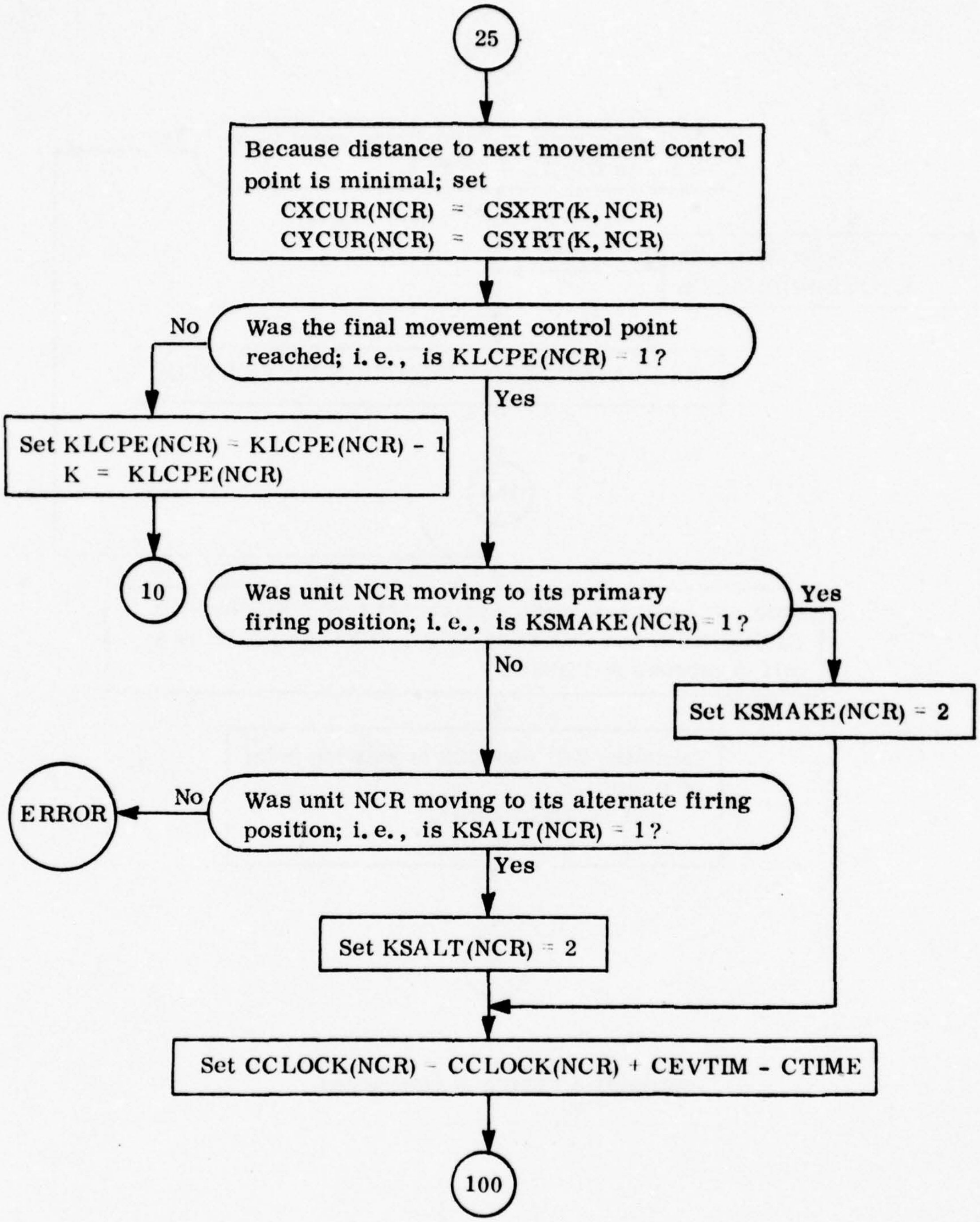
CSMOVE CALLED BY:

CSWCON	MOUNT

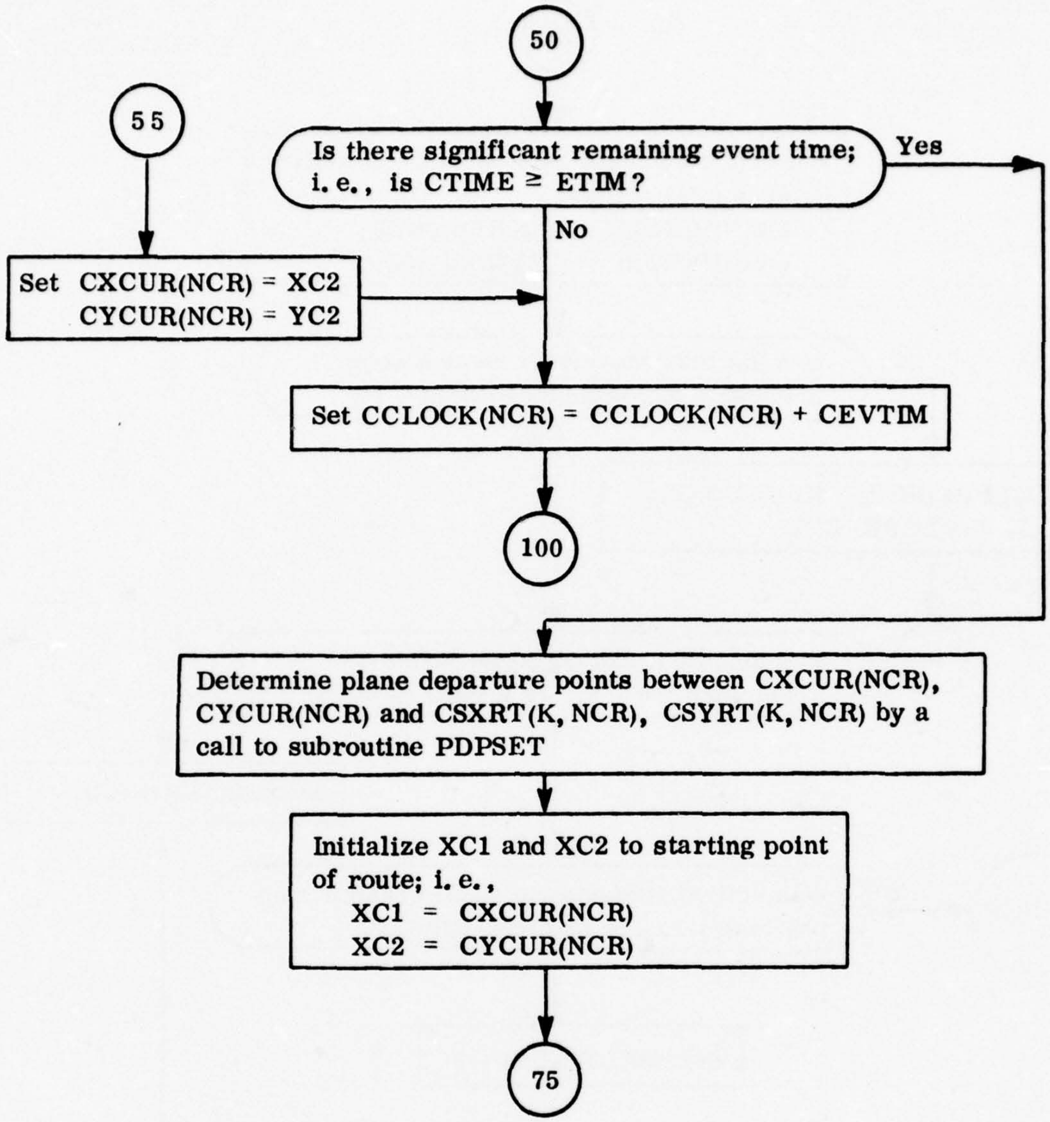
STORAGE: $DBA_{16} = 8514_{10}$ bytes



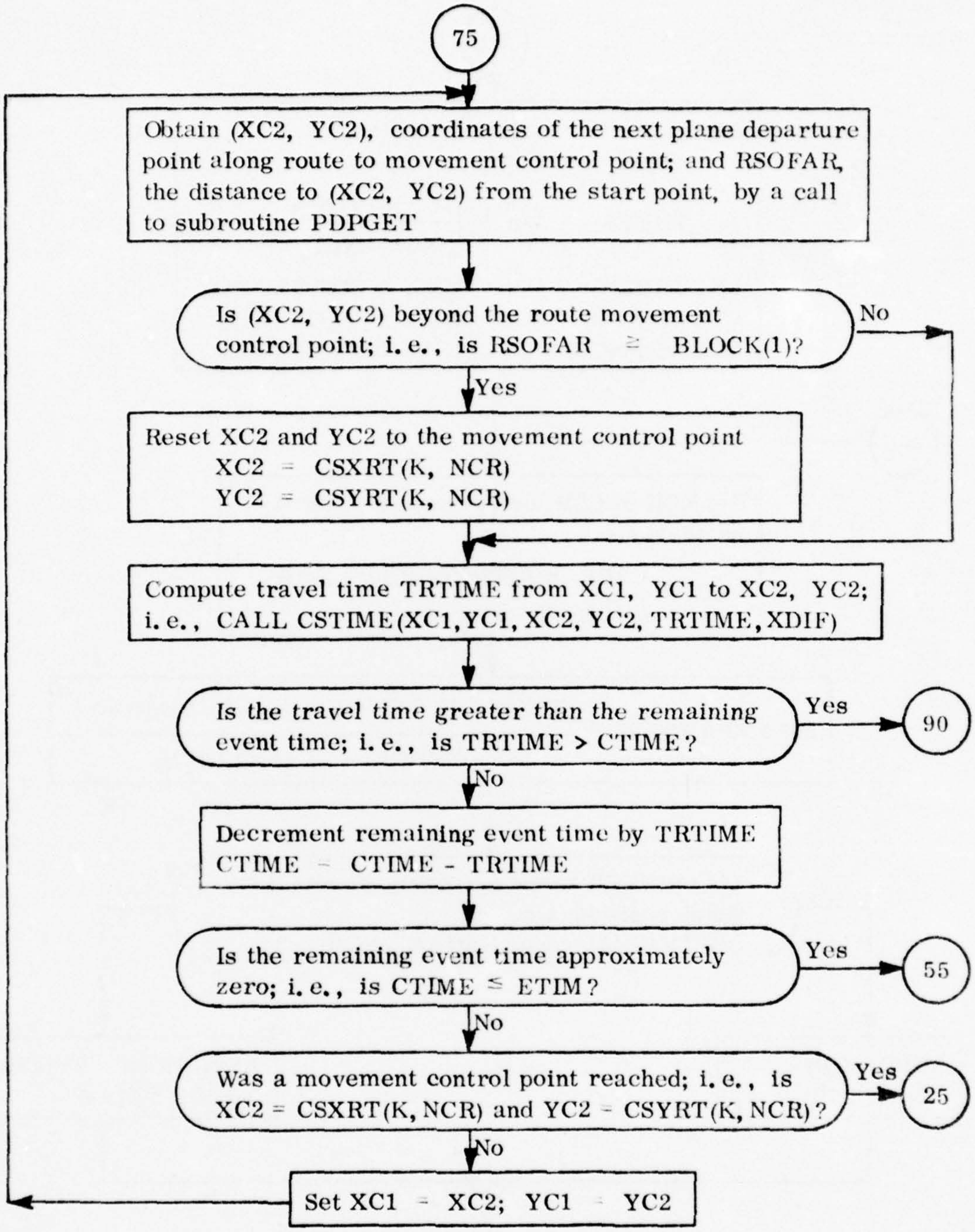
Subroutine CSMOVE: Crew-Served Weapon Movement Model
(Dismounted Mode)



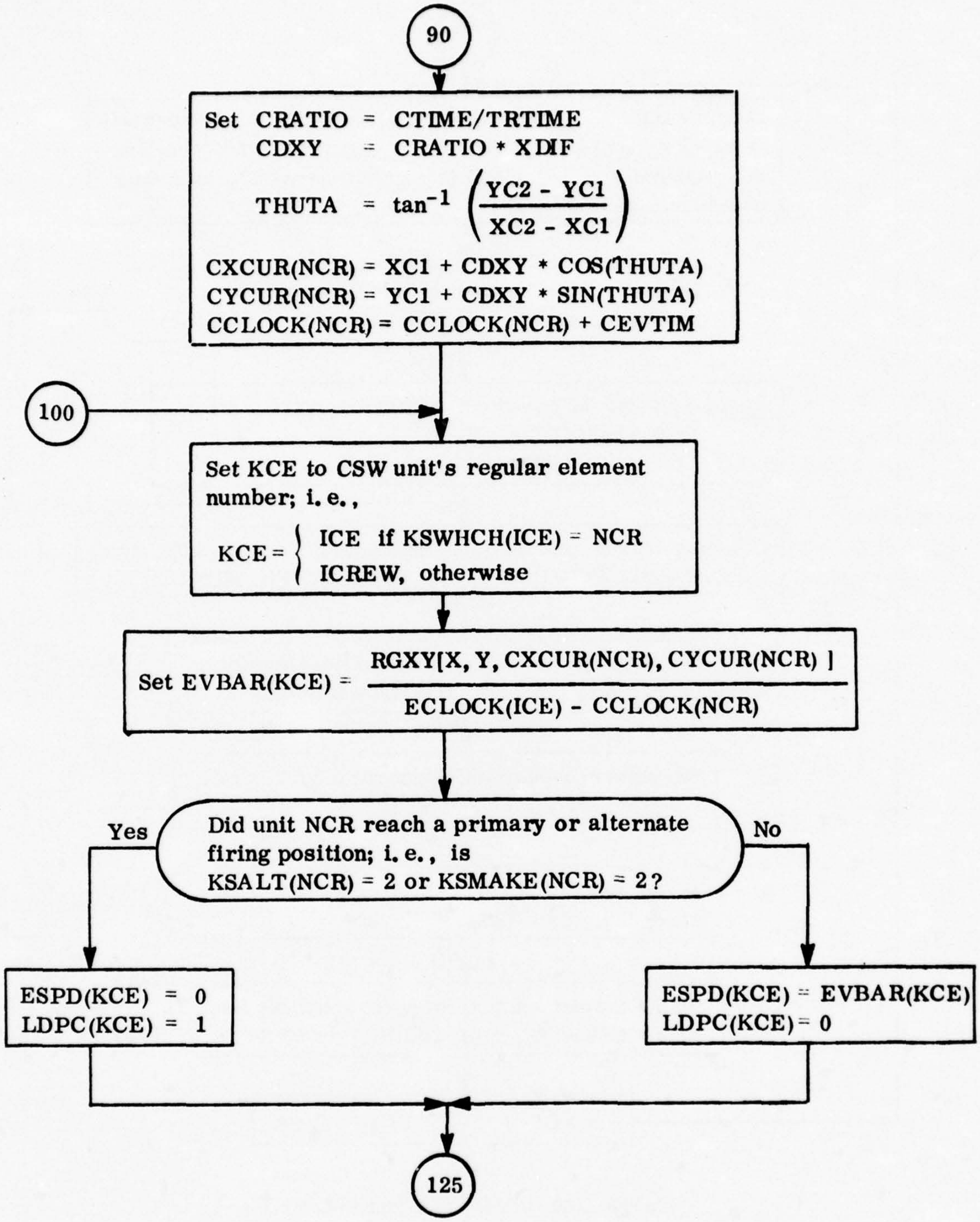
Subroutine CSMOVE: Continued



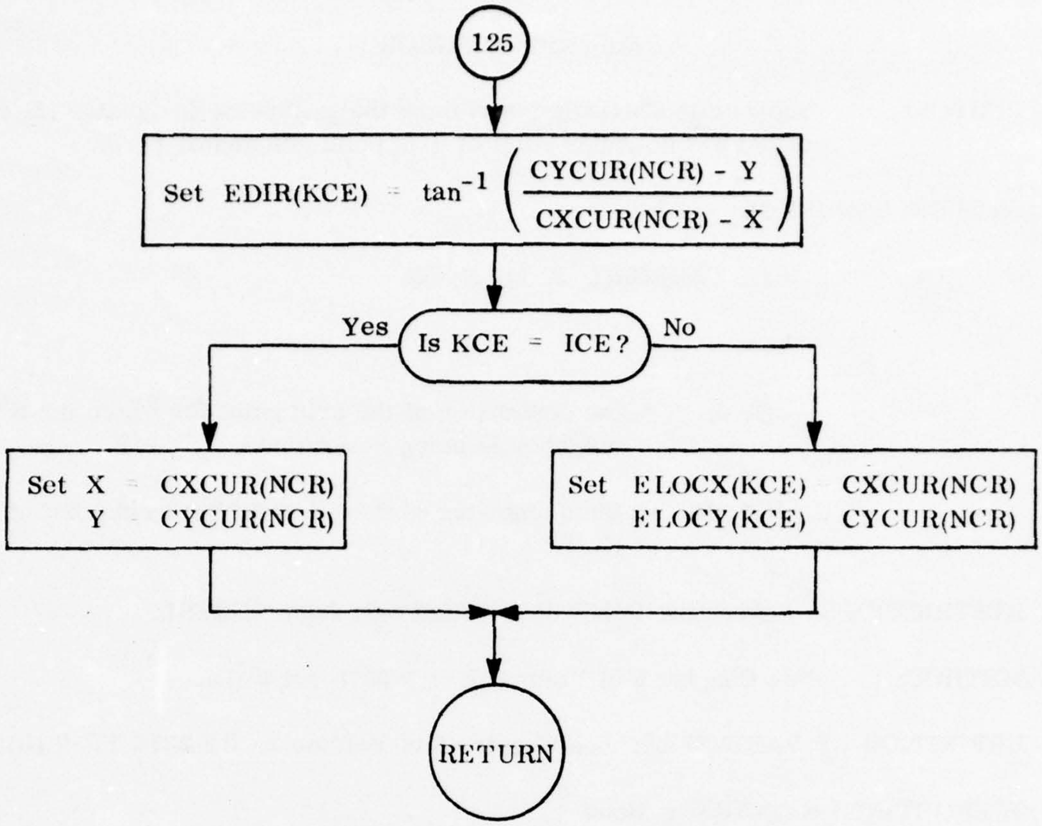
Subroutine CSMOVE: Continued



Subroutine CSMOVE: Continued



Subroutine CSMOVE: Continued



Subroutine CSMOVE: Continued

Subroutine CSNBOR

PURPOSE: Subroutine CSNBOR determines the grid point designator (M, N) of the K^{th} neighbor point of grid point designator (I, J).

CALLING SEQUENCE:

```
CALL CSNBOR(I, J, M, N, K)
```

where

(I, J) = the designator of the grid point for which the K^{th} neighbor is being determined;

(M, N) = the designator of the K^{th} neighbor grid point of (I, J).

RESTRICTIONS: Subroutine CSNBOR is called only from CSRTSL.

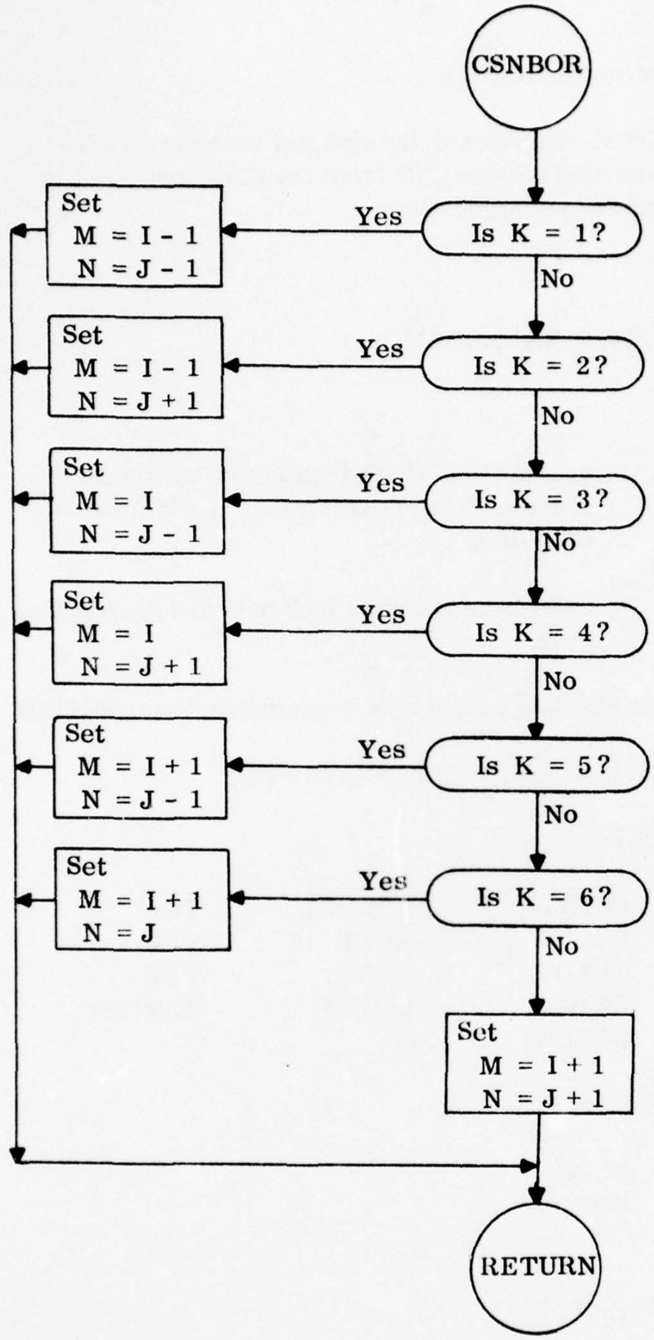
METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED: None

COMMON AREAS REFERENCED: None

STORAGE: $29A_{16} = 666_{10}$ bytes



Subroutine CSNBOR: Crew-Served Weapons--Route Selection Grid Neighbor Points

Subroutine CSRTSL

PURPOSE: Subroutine CSRTSL determines the optimal route for a dismounted crew-served weapon unit from the dismount point to the primary desired firing position.

CALLING SEQUENCE:

CALL CSRTSL(XDF, YDF, JRMAX)

where

(XDF, YDF) = coordinates of the location of the primary desired firing position for the dismounted CSW unit.

JRMAX = number of columns in fire position selection grid.

RESTRICTIONS: Subroutine CSRTSL is called only from subroutine CSWGRD.

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

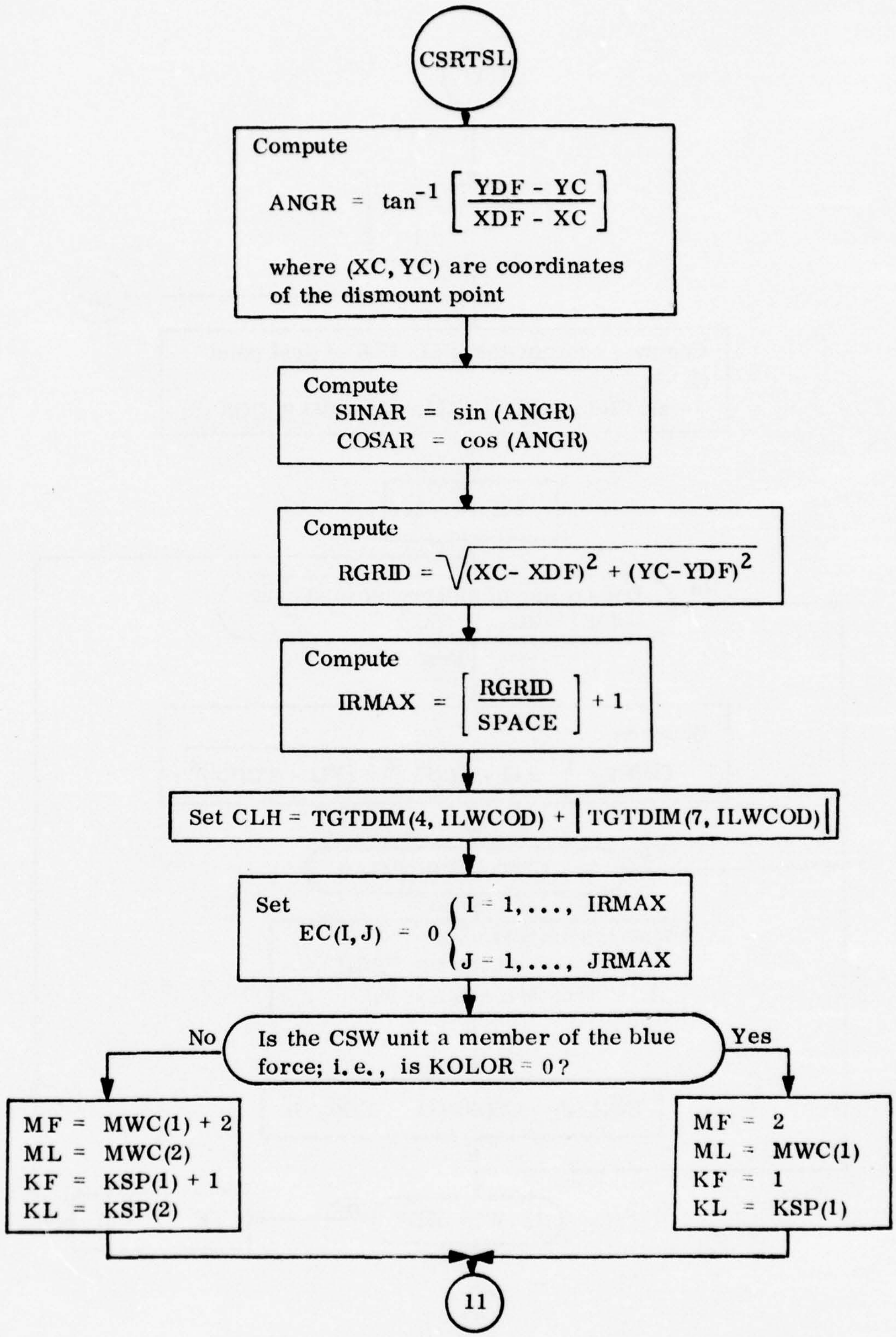
COMMON AREAS REFERENCED:

CSDEE	CSYRT	ICECOM	TC
CSDSP	EC	LDET	TEC
CSWDF1	ECLOCK	LKILL	TEW
CSWIND	EMICR	LWCOD	TGTDIM
CSXRT	HCREW	SC	

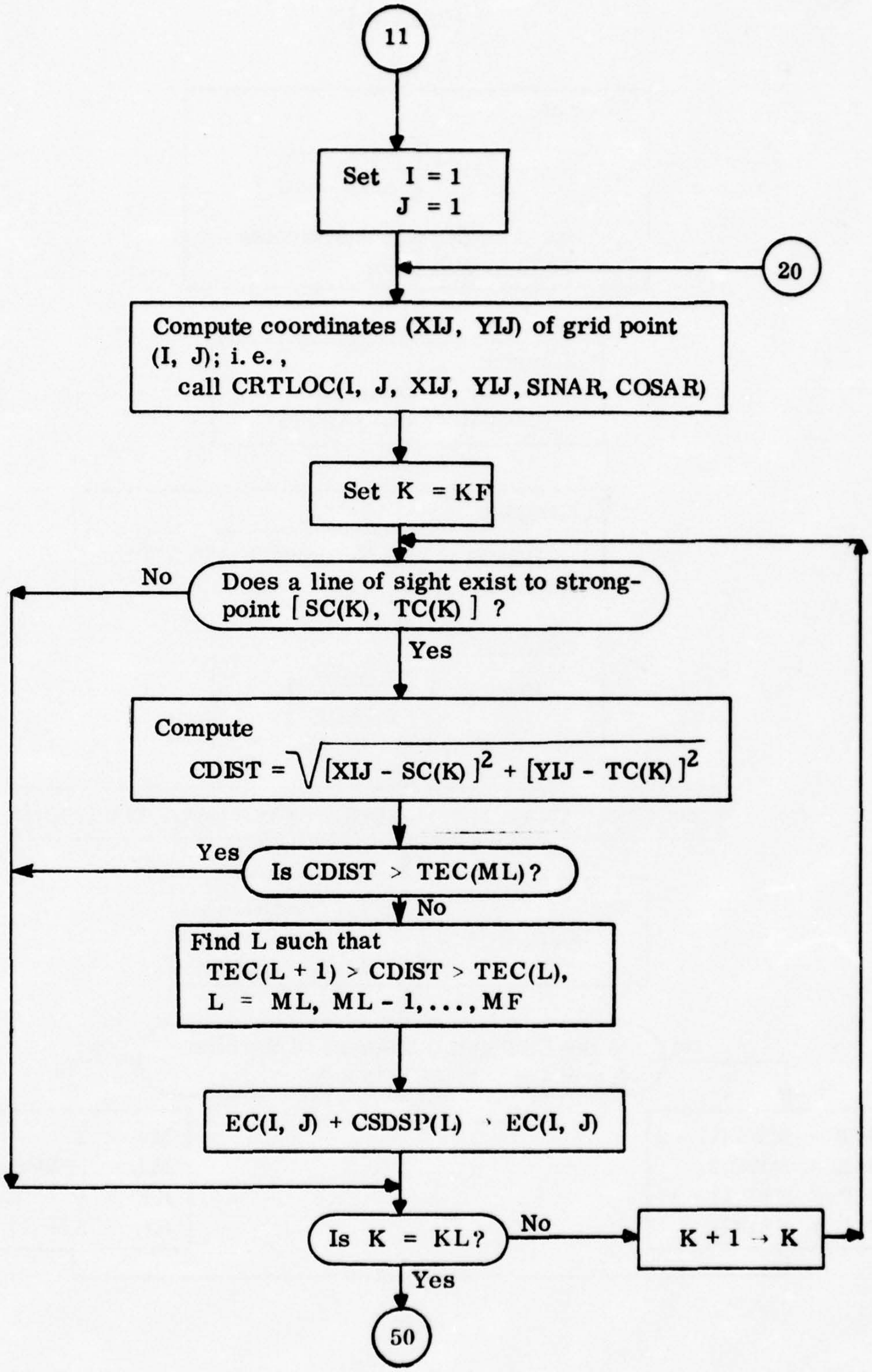
SUBROUTINES REQUIRED:

CRTLOC	ELOC
CSNBOR	LOSS
CSTCDF	

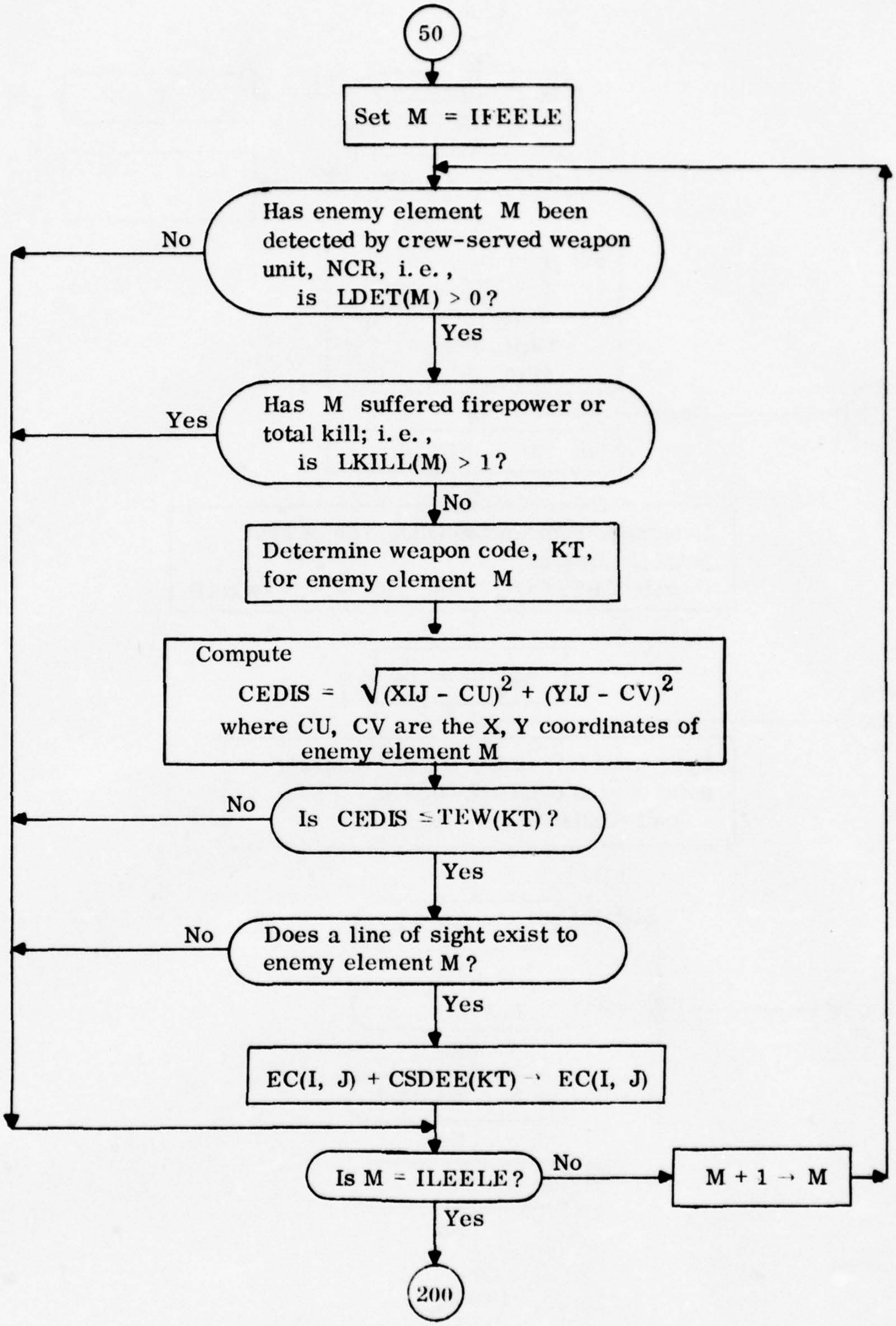
STORAGE: $38D4_{16} = 14548_{10}$ bytes



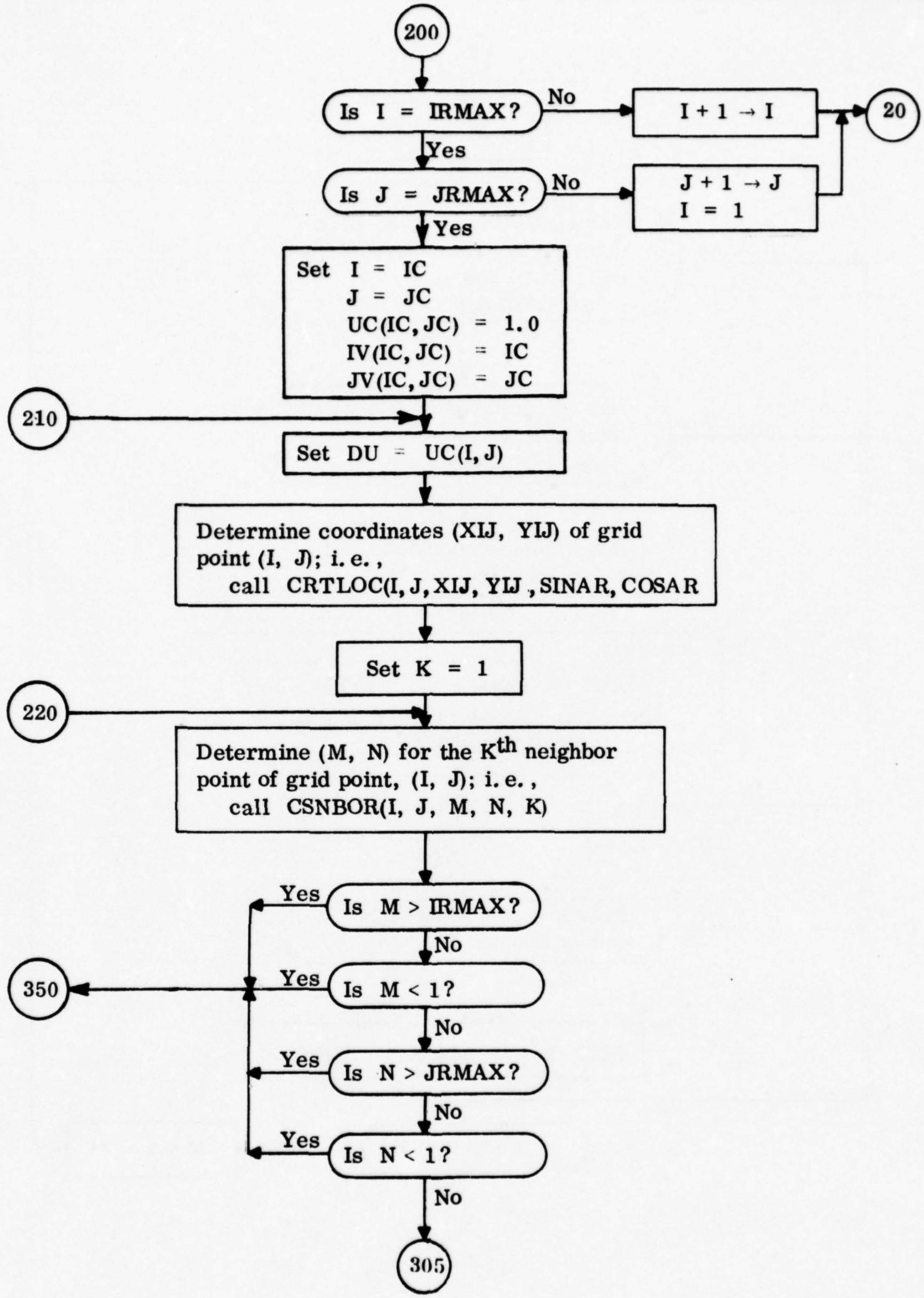
Subroutine CSRTSL: Crew-Served Weapons Route Selection Model (Dismounted Mode)



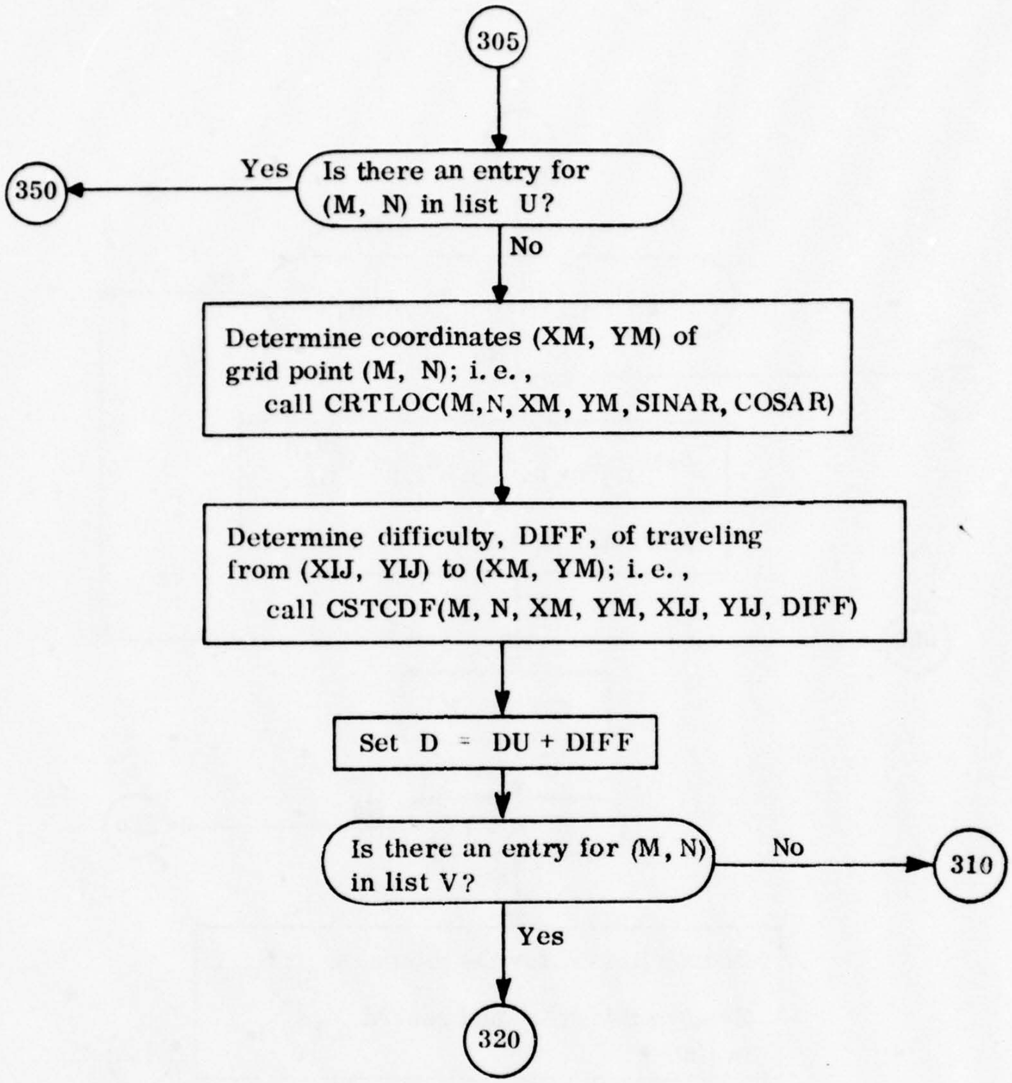
Subroutine CSRTSL: Continued



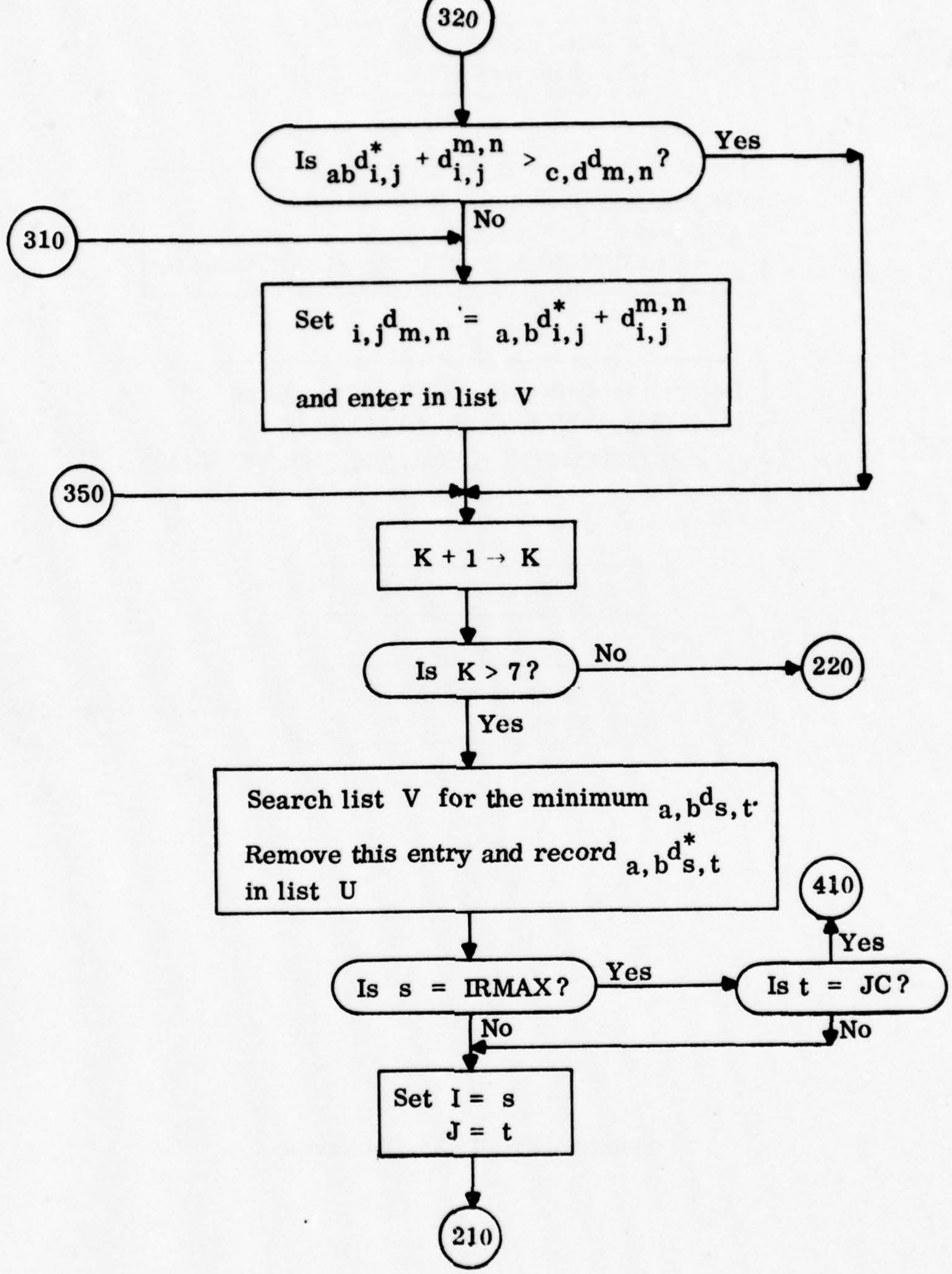
Subroutine CSRTSL: Continued



Subroutine CSRTSL: Continued



Subroutine CSRTSL: Continued



Subroutine CSRTSL: Continued

(410)

Compute [CSXRT(I, NCR), CSYRT(I, NCR)], the coordinates of the complete minimum difficulty route to grid point (IRMAX, JC);
where:

I =
- LAST ~ coordinates of the dismount point (XC, YC) for the CSW unit, corresponding to route selection grid point (IC, JC);
- ⋮
- 1 ~ coordinates of the final point of the route (XDF, YDF) corresponding to grid point (IRMAX, JC)

RETURN

Subroutine CSRTSL: Continued

Subroutine CSTCDF

PURPOSE: Subroutine CSTCDF determines the tactical difficulty of a route between two specified route selection grid points for a dismounted crew-served weapon unit.

CALLING SEQUENCE:

CALL CSTCDF(M, N, XM, YM, XIJ, YIJ, DIFF)

where

(M, N)	=	the designator of the route selection grid point to which the difficulty of the route is being computed;
(XM, YM)	=	the coordinates of grid point (M, N);
(XIJ, YIJ)	=	the coordinates of the grid point of the route selection grid from which the difficulty of the route is being computed;
DIFF	=	the tactical difficulty of the route from (XIJ, YIJ) to (XM, YM).

RESTRICTIONS: Subroutine CSTCDF is called only from subroutine CSRTSL.

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

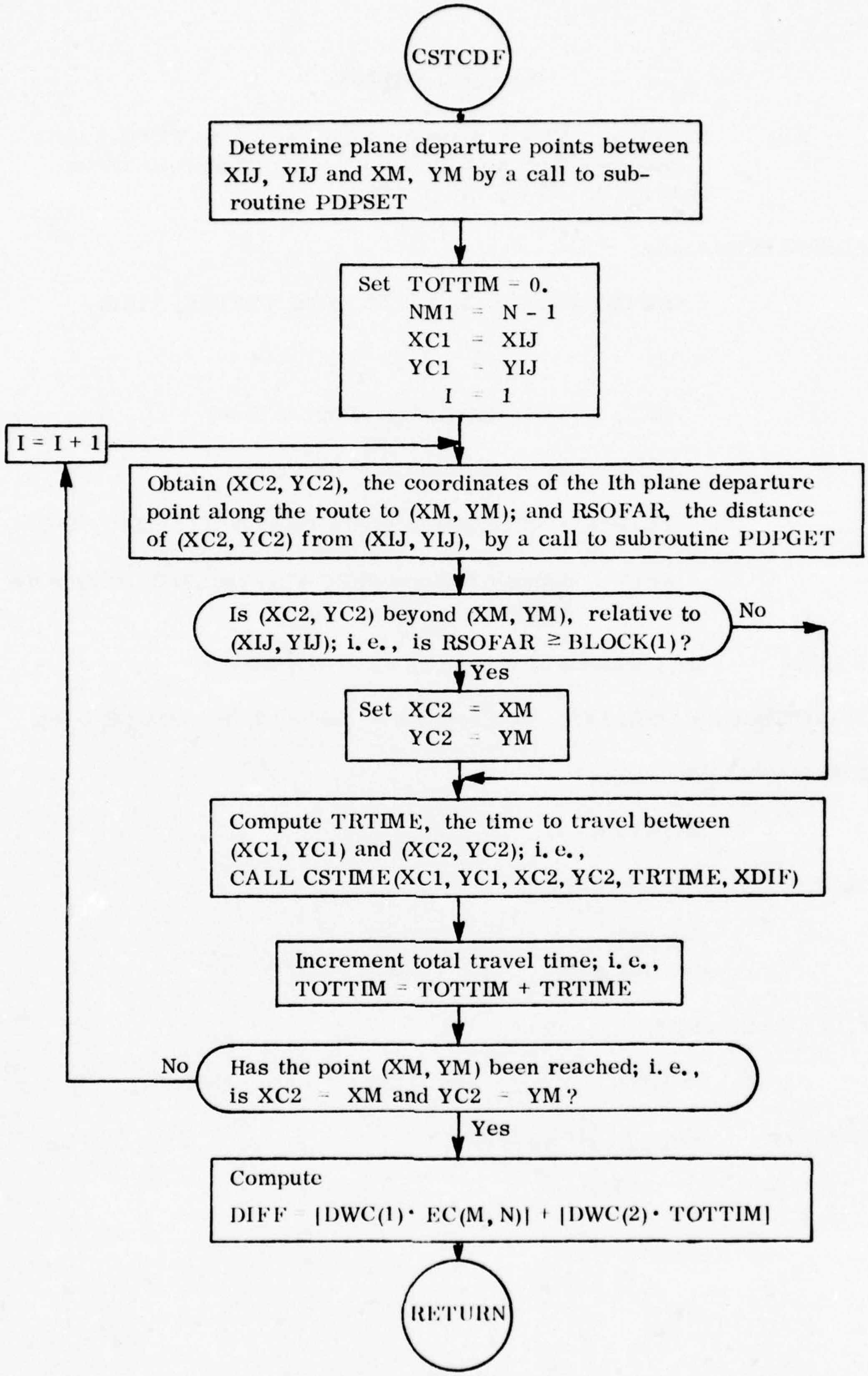
DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED:

CSTIME
PDPGET
PDPSET

COMMON AREAS REFERENCED: EC DWC

STORAGE: $382_{16} = 898_{10}$ bytes



Subroutine CSTCDF: Crew-Served Weapon Route Selection Difficulty

Subroutine CSTIME

PURPOSE: Subroutine CSTIME determines the travel time TRTIME for a dismounted CSW unit with weapon code, NKWEP, to travel from (XC1, YC1) to (XC2, YC2)

CALLING SEQUENCE:

CALL CSTIME(XC1, YC1, XC2, YC2, TRTIME, XDIF)

where

(XC1, YC1) = coordinates of initial point;

(XC2, YC2) = coordinates of final point

TRTIME = time to travel from (XC1, YC1) to (XC2, YC2)

XDIF = distance between (XC1, YC1) and (XC2, YC2) in the X-Y plane

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED:

ELVATE

COMMON AREAS REFERENCED:

CSWDF1	XCSWIN
CSWIND	YCSWIN

CSTIME CALLED BY:

CSMOVE	CSTIME	MOUNT

STORAGE: $45A_{16} = 1114_{10}$ bytes

CSTIME

Initialize travel time to zero; i. e., set TRTIME = 0

Compute elevation at (XC1, YC1), noted by HC1, and at (XC2, YC2), noted by HC2; i. e.,

HC1 = ELVATE (XC1, YC1, 0)

HC2 = ELVATE(XC2, YC2, 0)

Compute HDIF, the difference in elevations; i. e.,

$$HDIF = HC2 - HC1$$

Compute X-Y distance from (XC1, YC1) to (XC2, YC2); i. e.,

$$XDIF = \sqrt{(XC1 - XC2)^2 + (YC1 - YC2)^2}$$

Compute grade angle, THATA, from (XC1, YC1, HC1) to (XC2, YC2, HC2); i. e.,

$$THATA = \tan^{-1}\left[\frac{HDIF}{XDIF}\right]$$

Is angle THATA steeper than any grade angle in the XCSWIN array; i. e., is THATA ≥ XCSWIN(INCMAX)?

Yes → RETURN

No ↓

Is THATA in the range of the table; i. e., is THATA ≥ XCSWIN(1)?

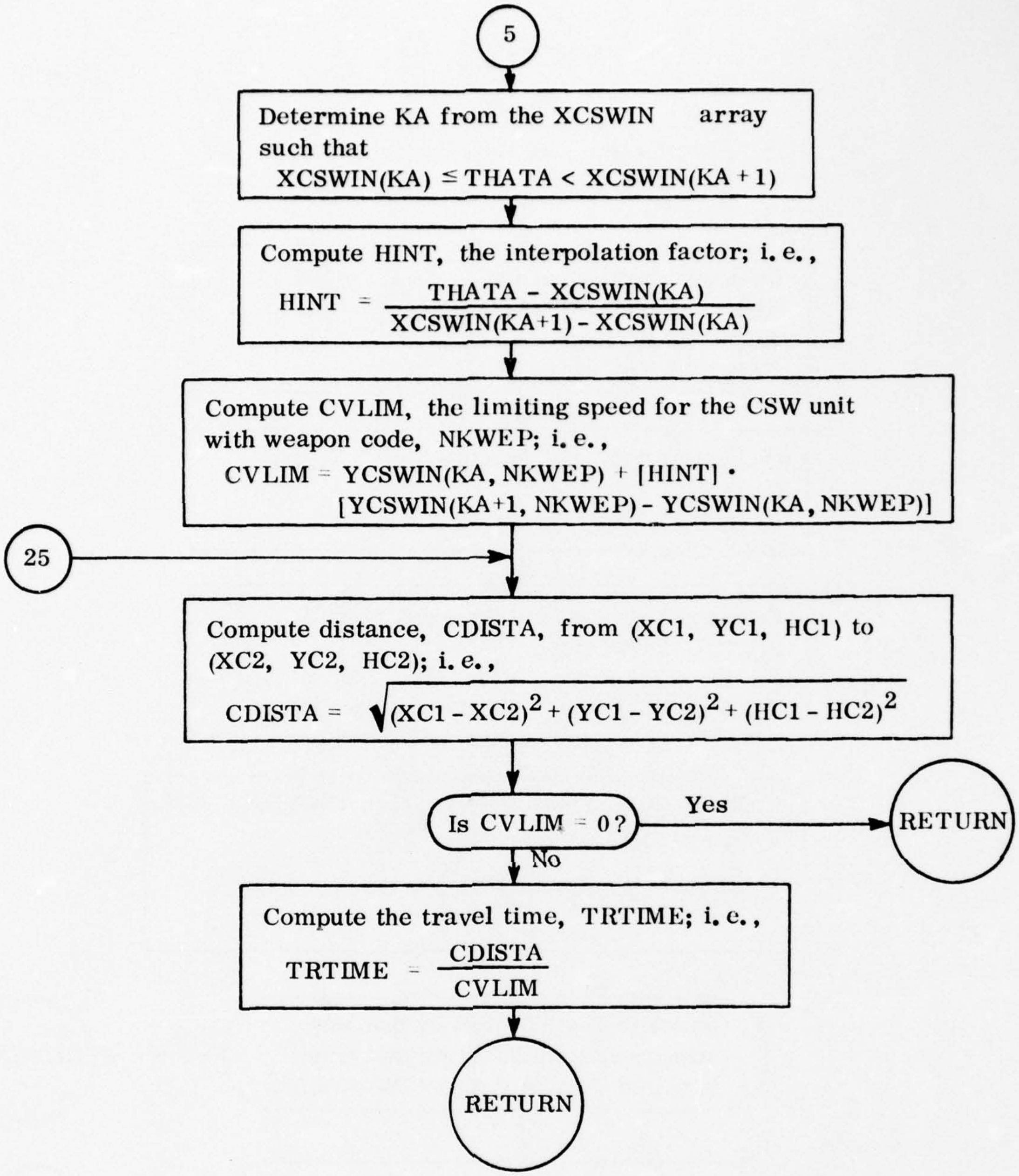
Yes → 5

No ↓

Use first limiting speed in table CVLIM = YCSWIN(1, NKWEP)

25

Subroutine CSTIME: Crew-Served Weapons--Travel Time Determination (Dismounted Mode)



Subroutine CSTIME: Continued

Subroutine CSWCAS

PURPOSE: To assess casualties against crew-served weapons and crew-served weapon carriers, and to update the corresponding force structures.

CALLING SEQUENCE:

CALL CSWCAS(ICAS,KILL)

where:

ICAS = casualty element number

$$
\text{KILL} = \begin{cases} 0, \text{ no kill} \\ 1, \text{ mobility kill} \\ 2, \text{ firepower kill} \\ 3, \text{ mobility and firepower kill} \\ 4, \text{ complete kill} \end{cases}
$$

DEFINITION OF VARIABLES:

LCSWFN(I) = crew-served weapon function code for DYNTACS element I, where:

$$
\text{LCSWFN(I)} = \begin{cases} 2 \text{ if element I is a CSW crew} \\ 1 \text{ if element I is a CSW carrier} \\ 0 \text{ otherwise} \end{cases}
$$

LCREW(I) = crew-served weapon status code, where:

$$
\text{LCREW(I)} = \begin{cases} 0 \sim \text{crews in section I are mounted} \\ 1 \sim \text{crews in section I are to dismount} \\ 2 \sim \text{crews in section I are dismounted} \\ 3 \sim \text{crews in section I are to mount} \end{cases}
$$

METHOD: See Chapter 9, Volume 1, RF 2376 FR 69-1 (U).

COMMON AREAS REFERENCED:

ECLOCK	KSWHCH	LSEC
ICECOM	LCREW	SEQPAR
ISORG	LCSWFN	TMNTD
KSDSMT	LKILL	

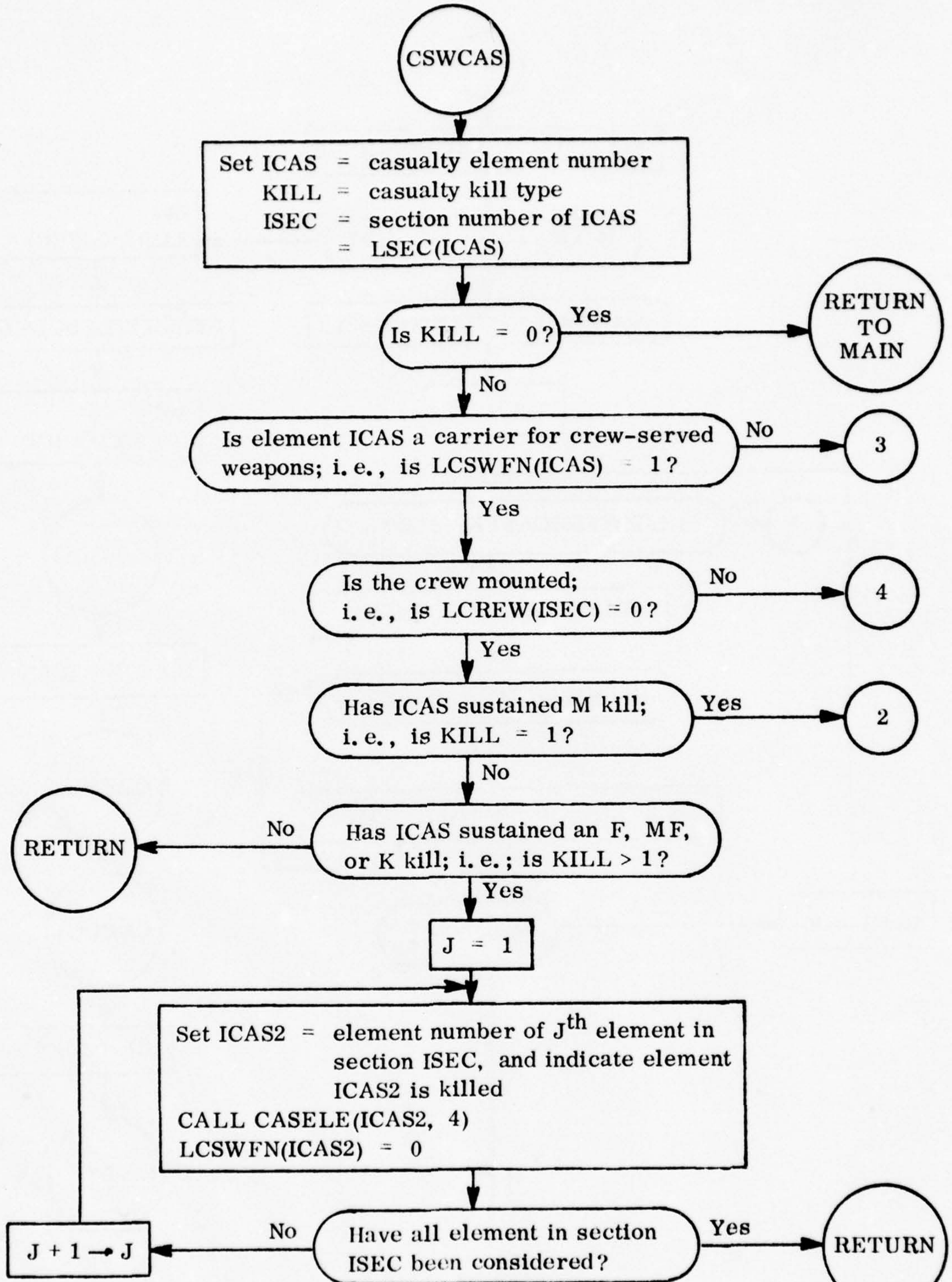
SUBROUTINES REQUIRED:

CASELE
CSWCON
GETICE
PUTICE

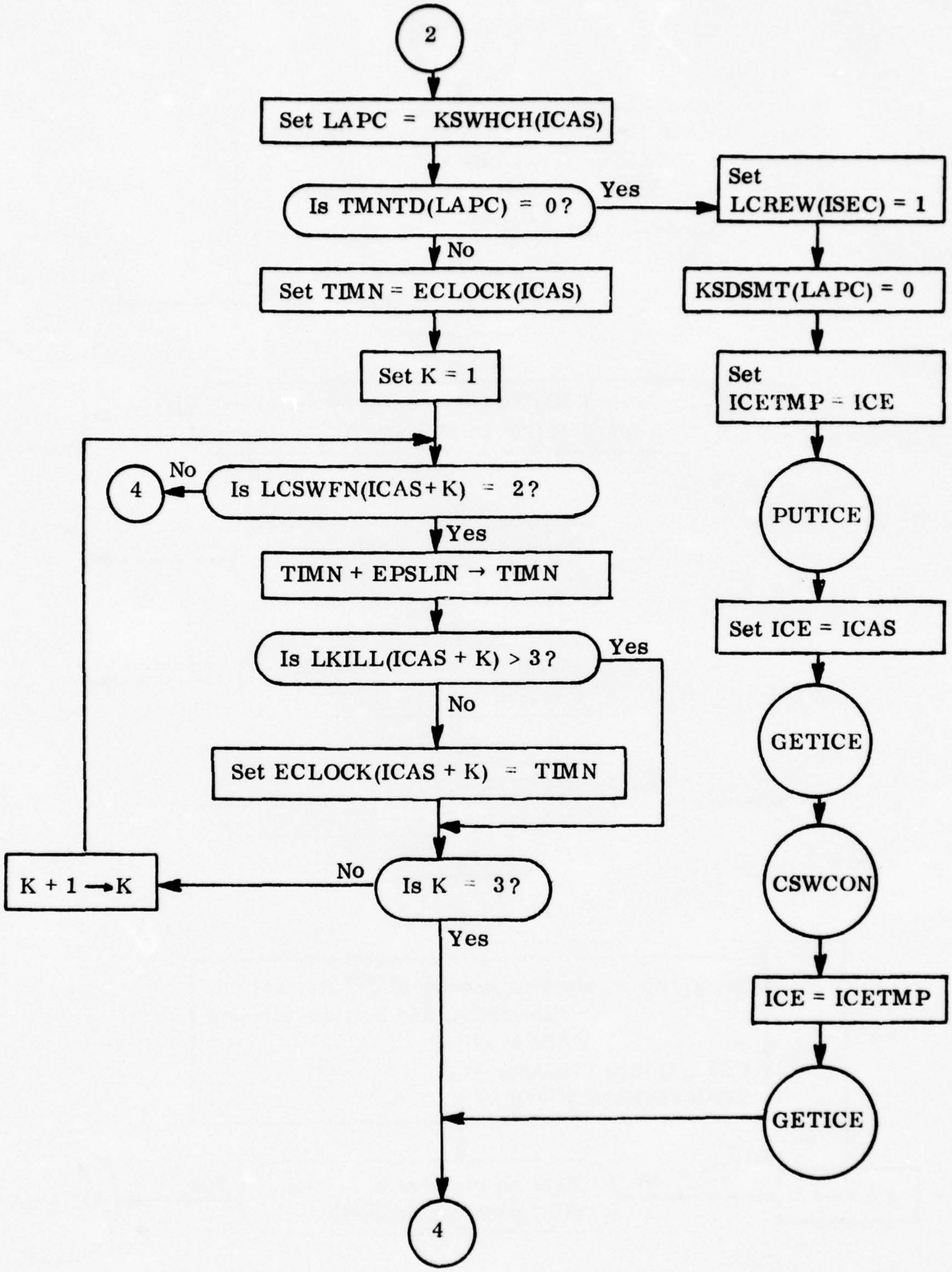
CSWCAS CALLED BY:

MAIN PROGRAM

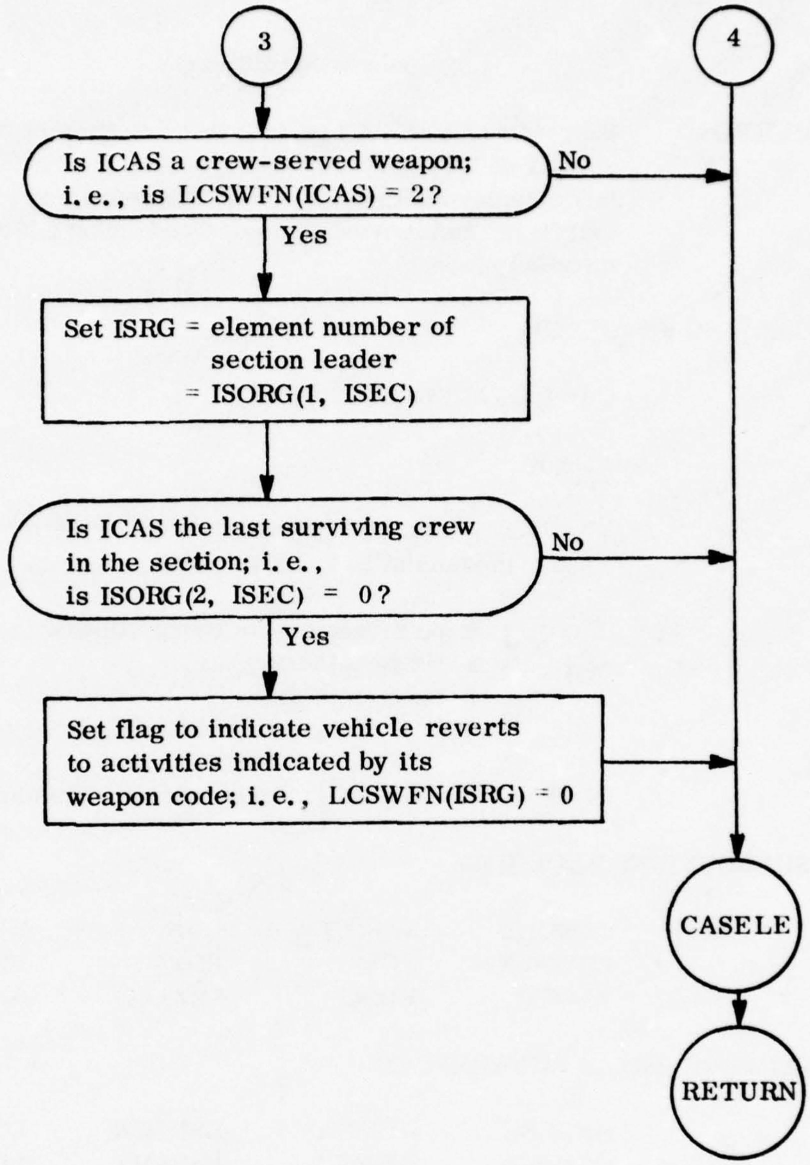
LENGTH: $514_{16} = 1300_{10}$ bytes



Subroutine CSWCAS: Assess Casualties



Subroutine CSWCAS: Continued



Subroutine CSWCAS: Continued

Subroutine CSWCON

PURPOSE: Subroutine CSWCON controls the activities of dismounted crew-served weapon units related to selection of attack or defense mode deployment, selection of desired firing positions, route selection, and movement to desired firing positions in the dismounted mode.

CALLING SEQUENCE:

CALL CSWCON (IGO, TIME)

where

IGO = flag returned indicating whether carrier firing and/or movement is to be allowed.

$$IGO = \begin{cases} 0 & \text{no movement or firing allowed} \\ 1 & \text{firing allowed} \\ 2 & \text{movement allowed} \\ 3 & \text{both firing and movement allowed} \end{cases}$$

TIME = *time increment to be added to the carrier's clock*

SUBROUTINES REQUIRED:

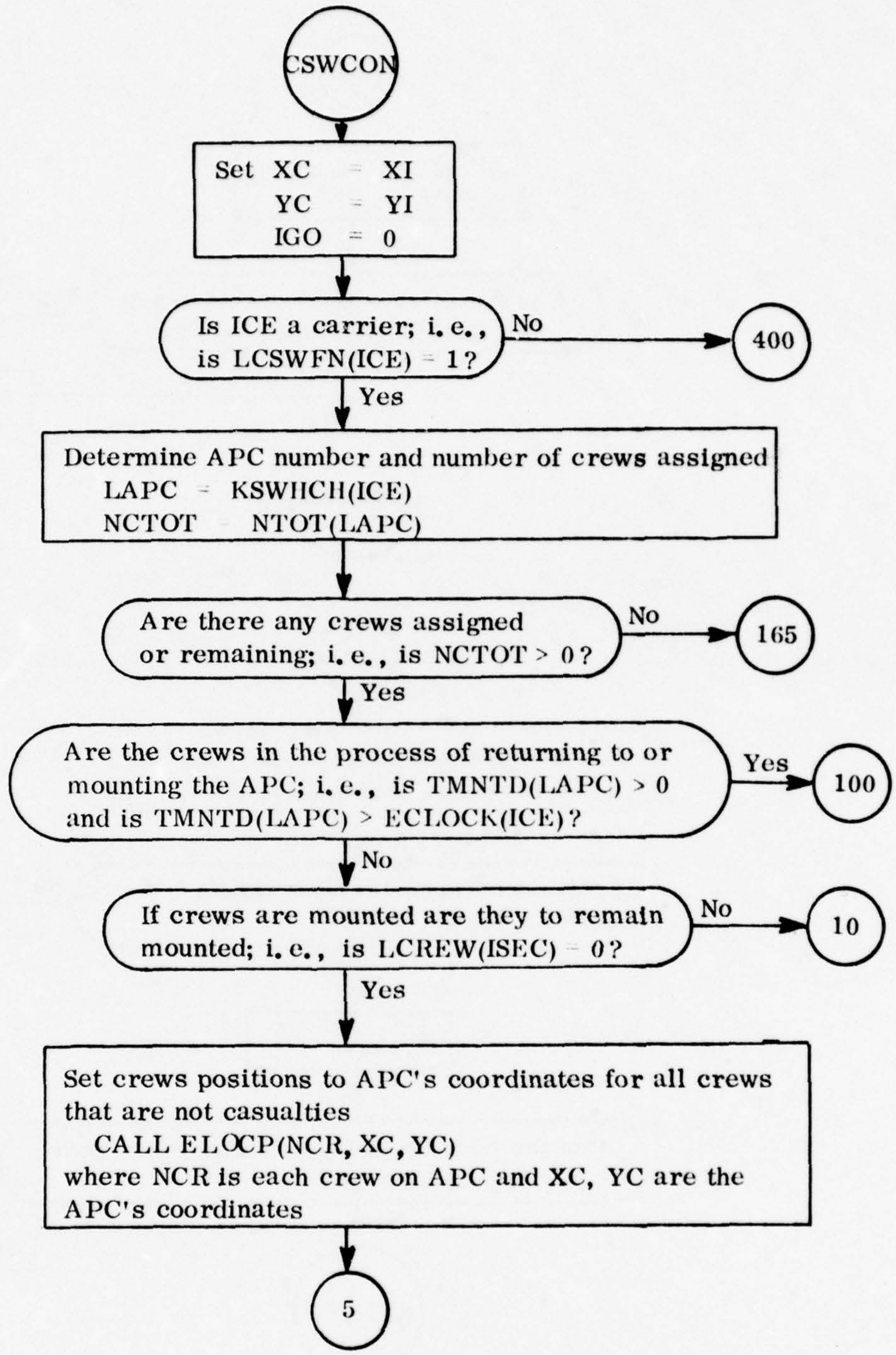
CSWGRD	MOUNT	CON	IDET
CSMOVE	COV	RGXY	ERROR
ELOCP	ELOC	SETLOS	IDETP

COMMON AREAS REFERENCED:

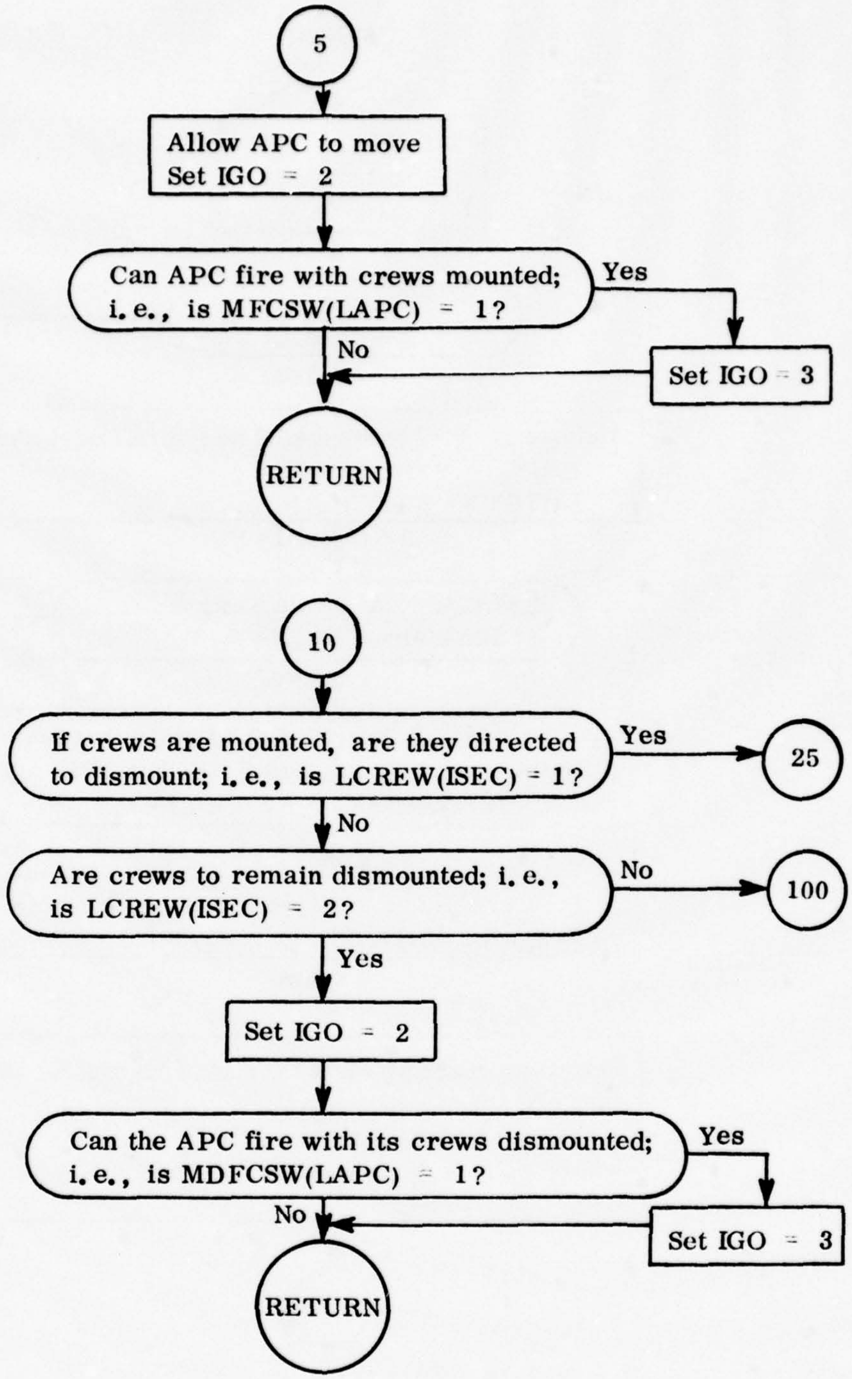
ANGSEC	DIRMU	KSDECN	LCSWTP	RMIN
CCLOCK	DISMTT	KSDSMT	LDET	NRDCNT
CSWDF1	ECLOCK	KSMAKE	LDPC	TMNTD
CSWIND	ELOCX	KSMVT	LKILL	XDF
CSXRT	ELOCY	KSWHCH	LTARG	YDF
CSYRT	ICECOM	LAPCCW	MDFCSW	XDFA
CXCUR	KLCPE	LCREW	MFCSW	YDFA
CYCUR	KSALT	LCSWFN	NTOT	

CSWCON CALLED FROM: CSWCAS, MAIN

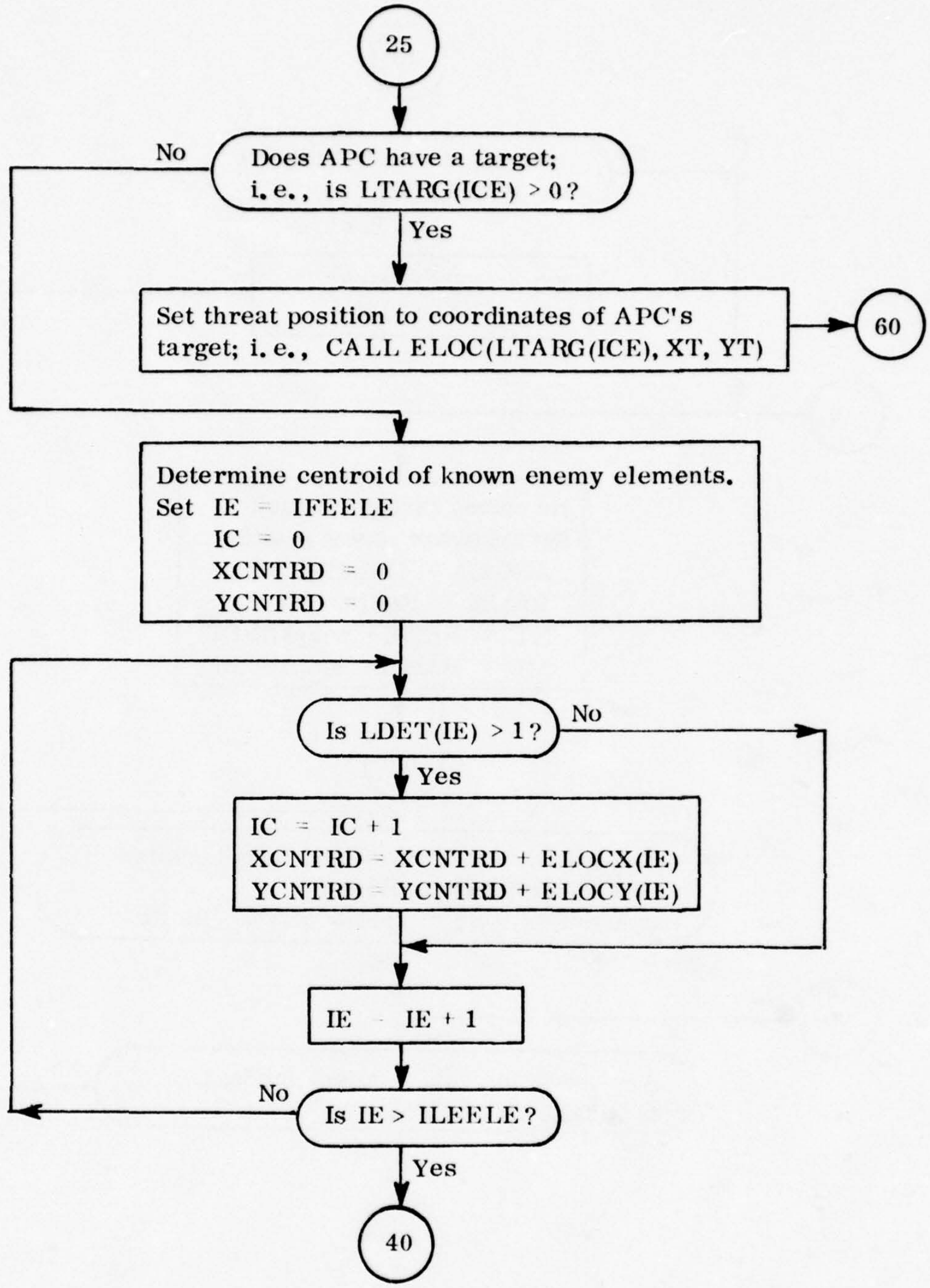
STORAGE: 1794_{16} = $\underline{6036}_{10}$ bytes



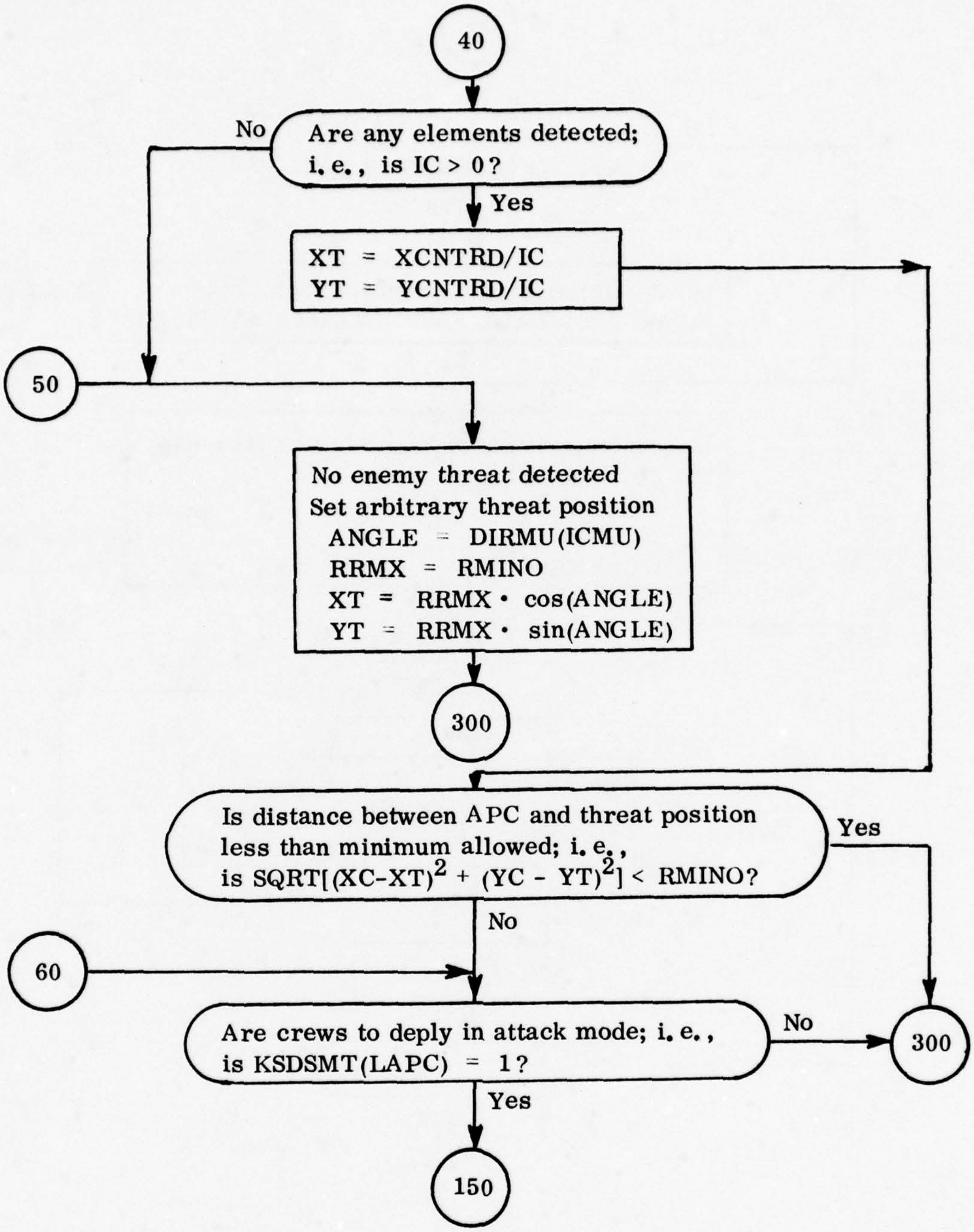
Subroutine CSWCON: Crew-Served Weapon Activity Controller
(Dismounted Mode)



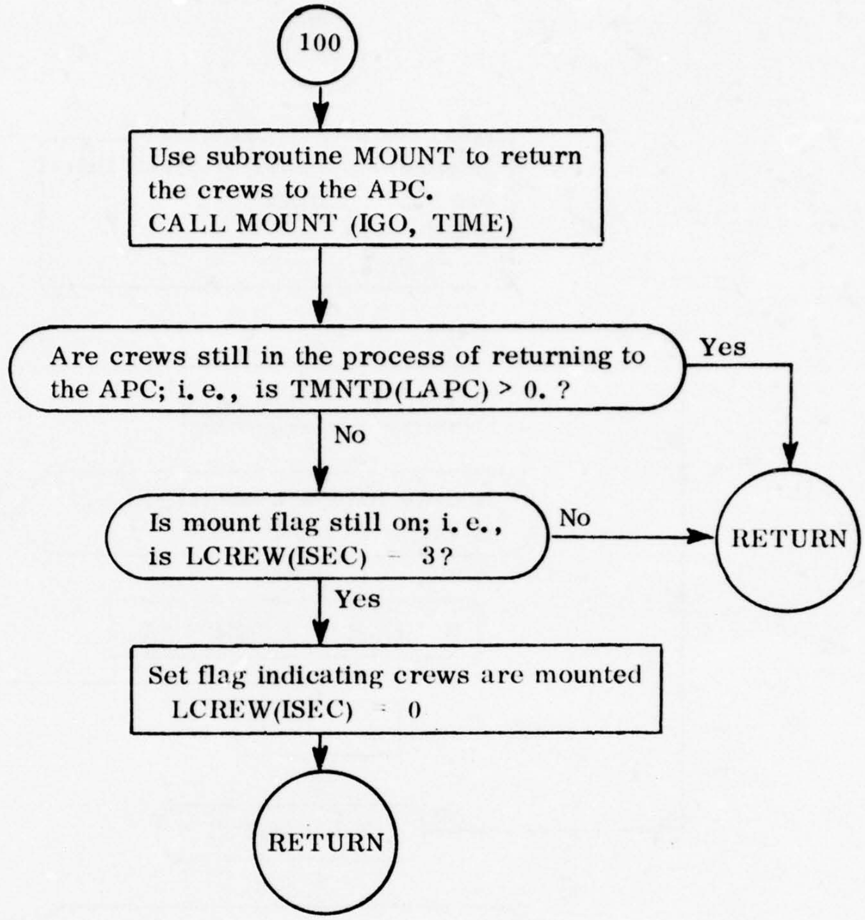
Subroutine CSWCON: Continued



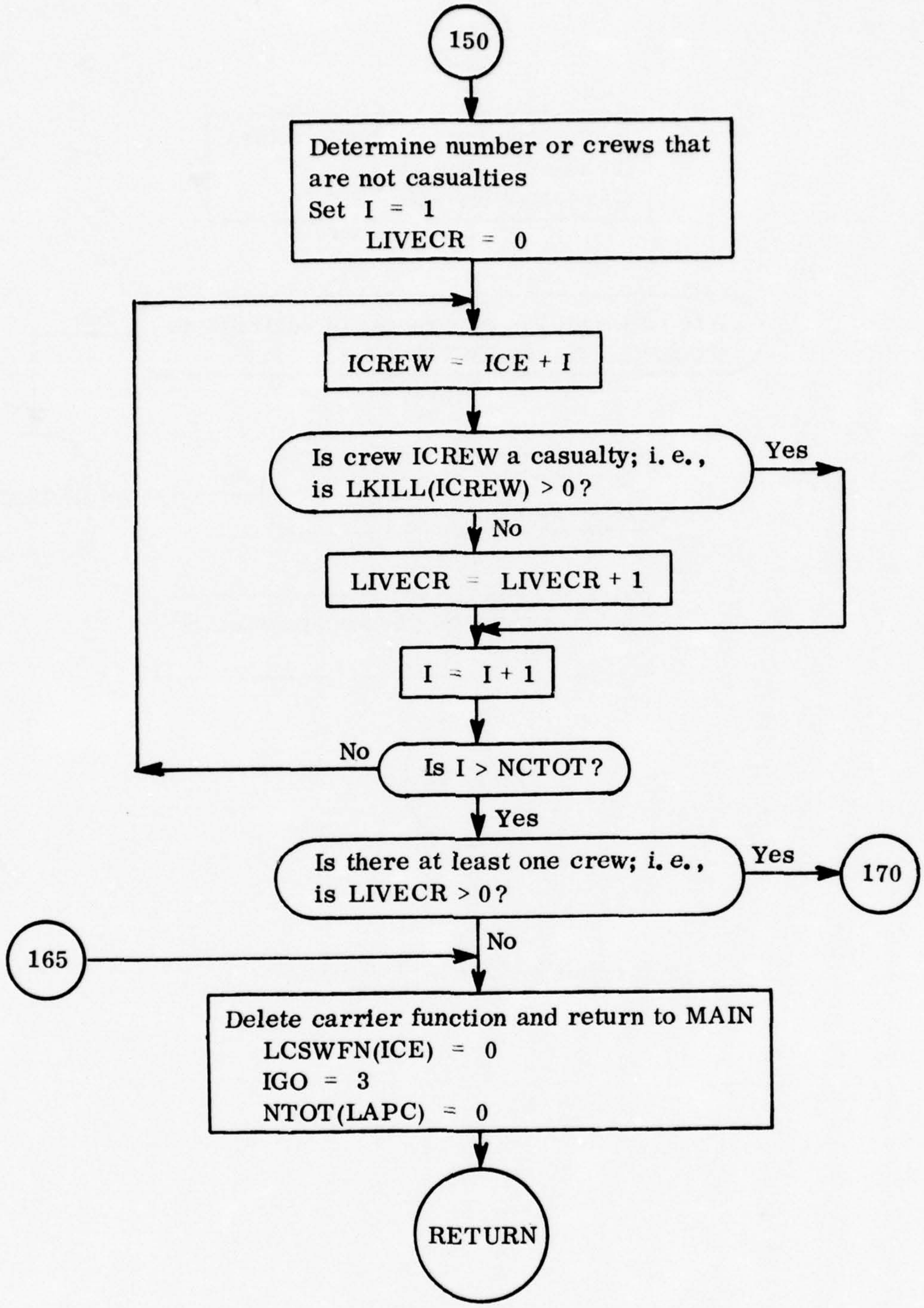
Subroutine CSWCON: Continued



Subroutine CSWCON: Continued



Subroutine CSWCON: Continued



Subroutine CSWCON: Continued

170

Is the sector of responsibility, ANGSEC(LAPC), for carrier, LAPC, non-negative; ;i. e., is ANGSEC(LAPC) $\geq$ 0?

No → ERROR

Yes

Compute

$$\text{SECTOR} = \frac{\text{ANGSEC(LAPC)}}{\text{LIVECR}}$$

Compute

$$\text{ANGMID} = \text{TAN}^{-1}\left(\frac{\text{YT} - \text{YC}}{\text{XT} - \text{XC}}\right)$$

$$\text{SECMIN} = \text{ANGMID} - \frac{\text{ANGSEC(LAPC)}}{2}$$

$$\text{SECMAX} = \text{ANGMID} + \frac{\text{ANGSEC(LAPC)}}{2}$$

Compute

$$\text{ANGAL} = \text{SECMIN} + \frac{\text{SECTOR}}{2}$$

Compute DTHRT, the distance to the threat position, (XT, YT), for carrier, LAPC; i. e.,

$$\text{DTHRT} = \sqrt{(\text{XC} - \text{XT})^2 + (\text{YC} - \text{YT})^2}$$

Set I = 1

12

Sburoutine CSWCON: Continued

12

Has crew I been killed; i. e., is LKILL(ICE + I) > 0?

Yes → 200

No

Determine CSW unit and weapon code numbers of crew I on LAPC

NCR = LAPCCW(I, LAPC)

NKWEP = LCSWTP(NCR)

Compute coordinates, (XTI, YTI), of threat position relative to which crew NCR will select its firing position, and the limiting distance, DIWC, for the firing position; i. e.,

$$XTI = XC + (DTHRT) \cdot \cos (ANGAL)$$

$$YTI = YC + (DTHRT) \cdot \sin (ANGAL)$$

$$DIWC = DTHRT - RMIN(NCR)$$

Compute coordinates, (XXR1, YYR1), and (XXR2, YYR2), of the boundary points for width of fire position selection grid; i. e.,

$$XXR1 = XC + [DIWC] \cdot \cos \left(ANGAL - \frac{SECTOR}{2}\right)$$

$$YYR1 = YC + [DIWC] \cdot \sin \left(ANGAL - \frac{SECTOR}{2}\right)$$

$$XXR2 = XC + [DIWC] \cdot \cos \left(ANGAL + \frac{SECTOR}{2}\right)$$

$$YYR2 = YC + [DIWC] \cdot \sin \left(ANGAL + \frac{SECTOR}{2}\right)$$

Compute width of fire position selection grid

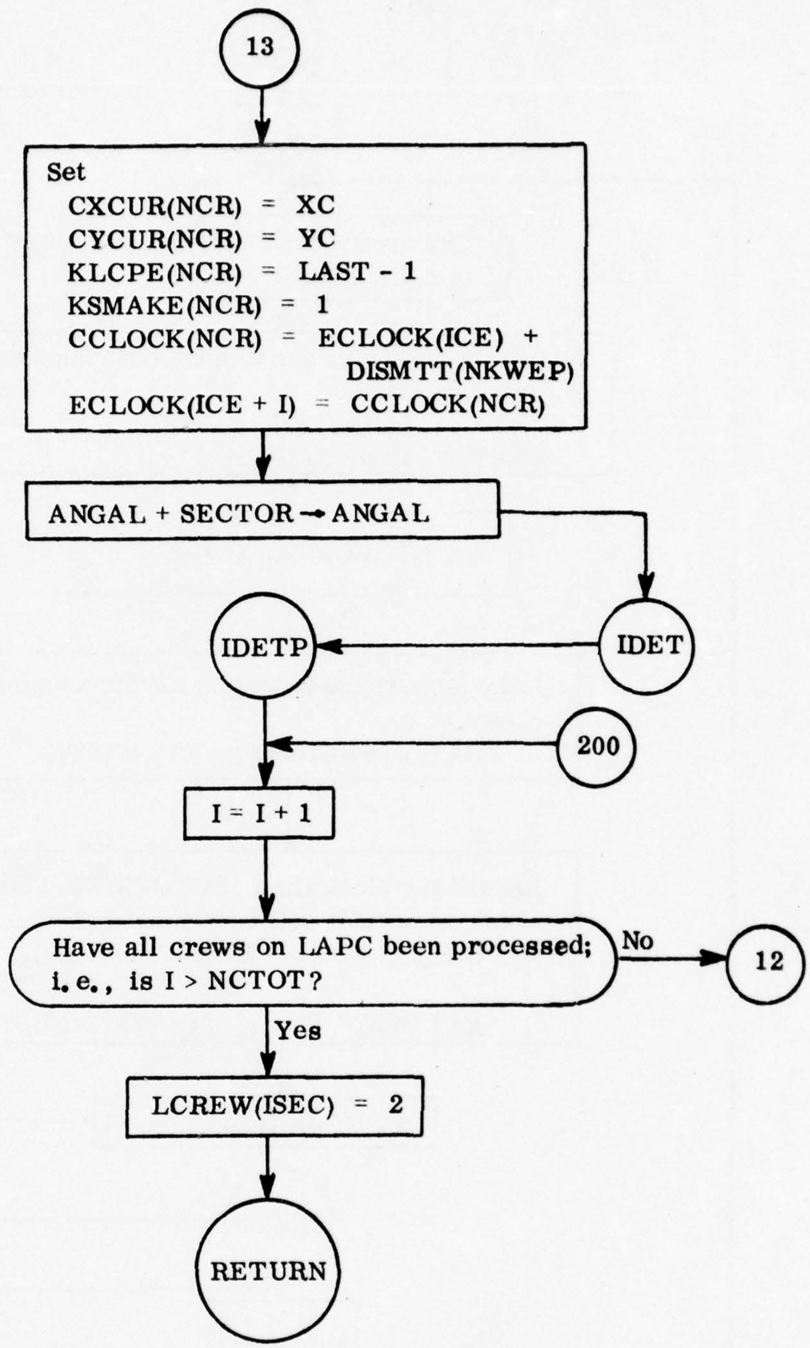
$$WIDTH = \sqrt{(XXR1 - XXR2)^2 + (YYR1 - YYR2)^2}$$

Determine firing position (XDF, YDF), alternate firing position (XDFA, YDFA), and the optimal route [CSXRT(I, NCR), CSYRT(I, NCR)] for crew NCR, i. e.,

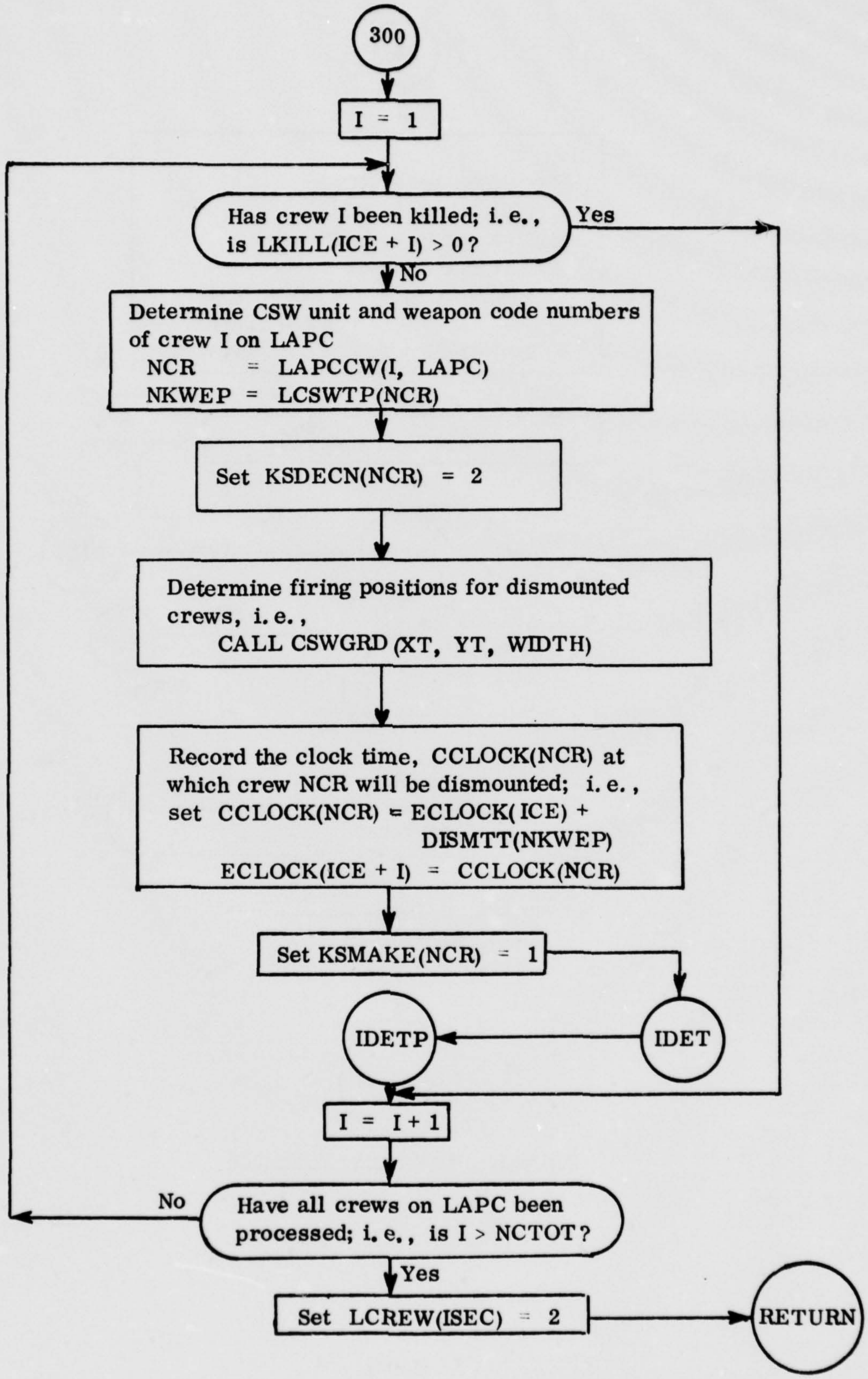
CALL CSWGRD (XTI, YTI, WIDTH)

13

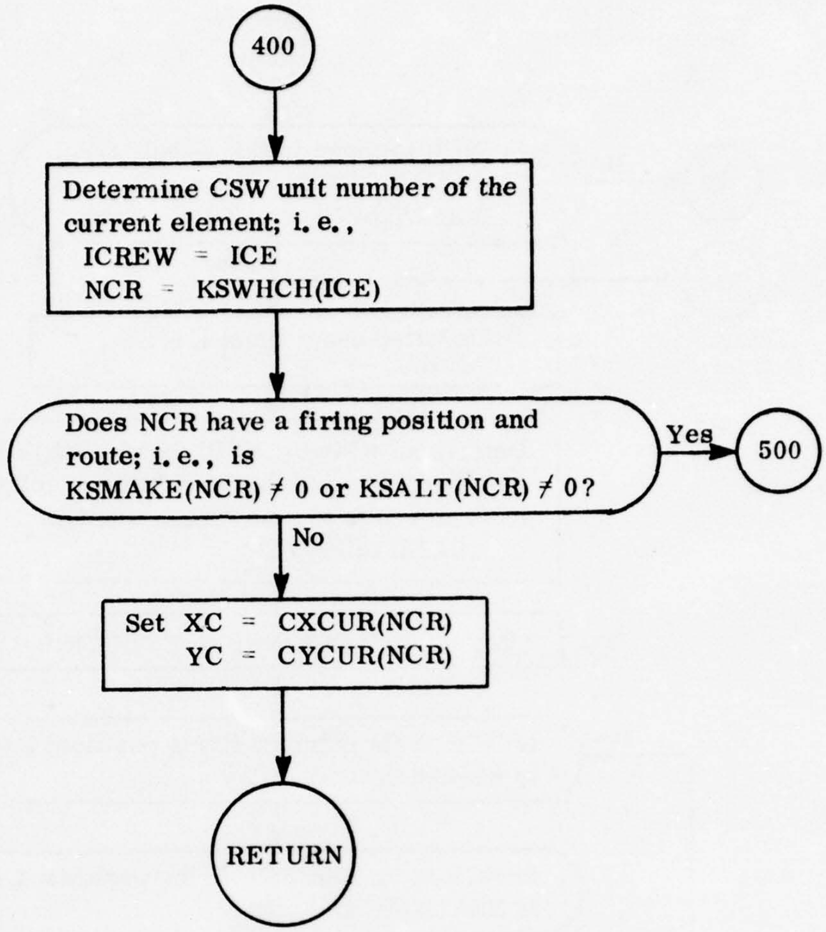
Subroutine CSWCON: Continued



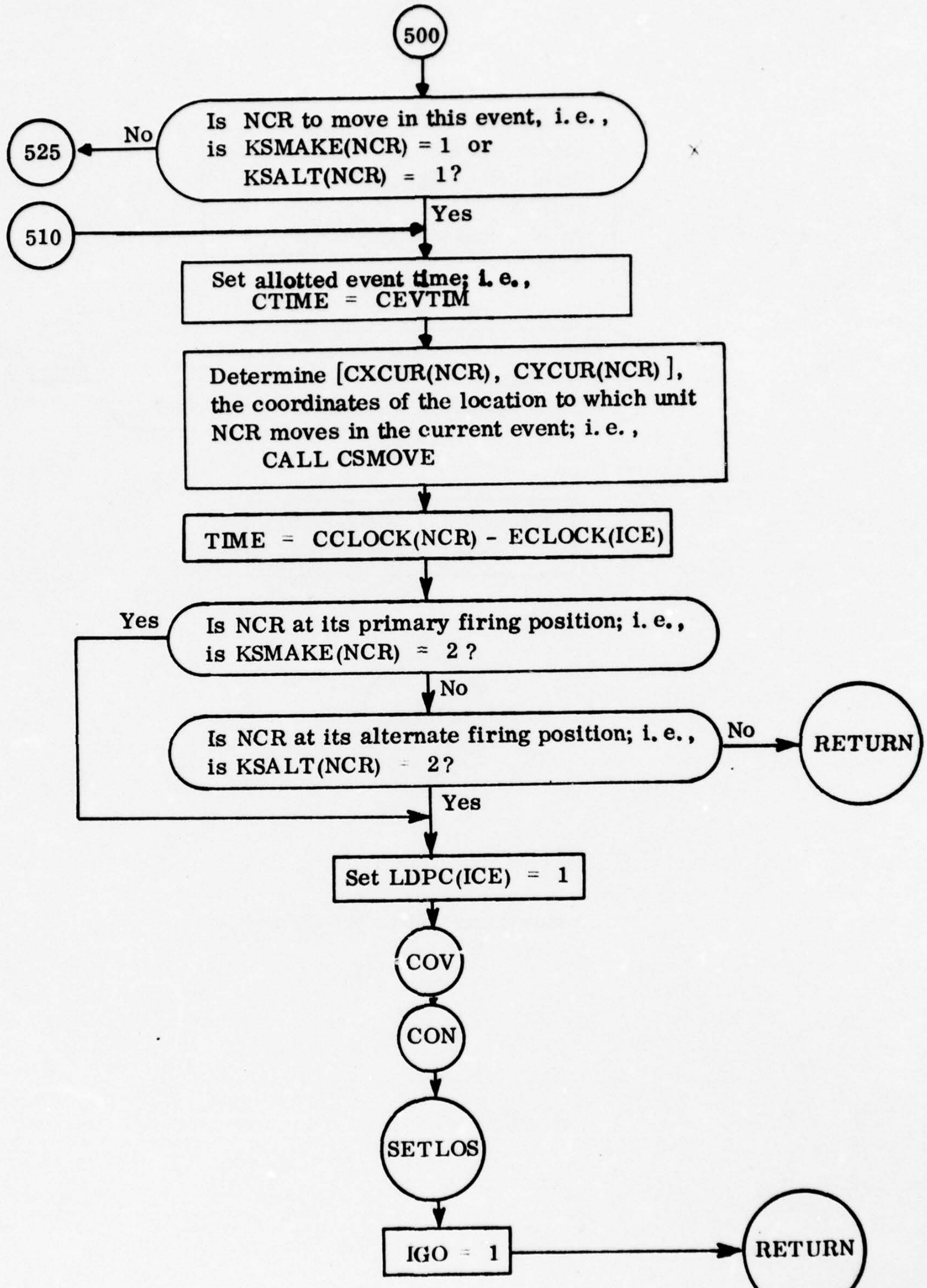
Subroutine CSWCON: Continued



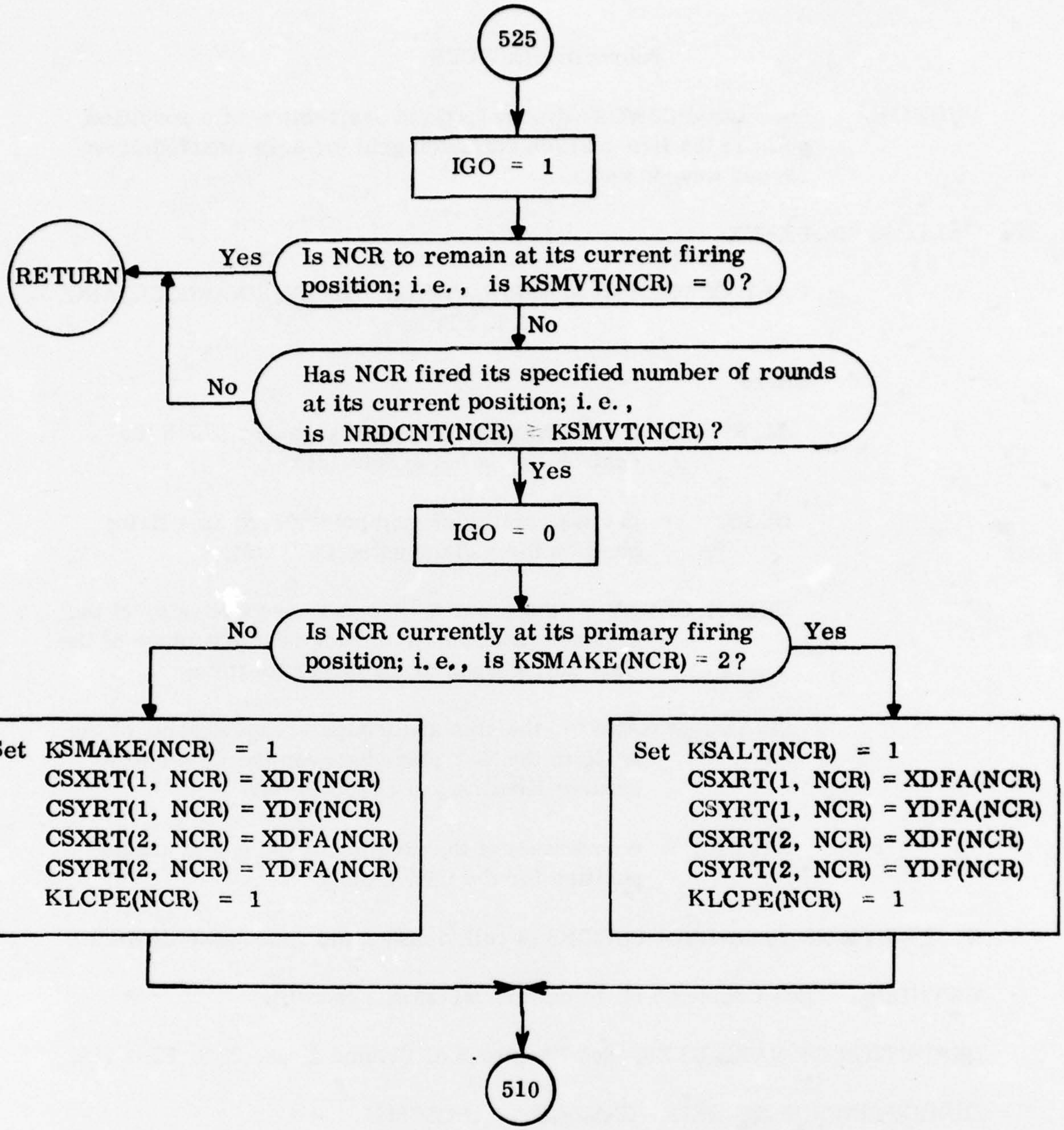
Subroutine CSWCON: Continued



Subroutine CSWCON: Continued



Subroutine CSWCON: Continued



Subroutine CSWCON: Continued

Subroutine CSWDES

PURPOSE: Subroutine CSWDES determines the desirability of a specified point in the fire position selection grid for a dismounted crew-served weapon unit.

CALLING SEQUENCE:

CALL CSWDES(M, N, DESIR, CDELX, CDELY, SINANG, COSANG, XT, YT)

where

(M, N) = the designator of the grid point for which the desirability is being determined;

DESIR = the desirability of grid point (M, N) as a firing position for a dismounted CSW unit.

CDELX, CDELY = the X and Y distance, respectively, of the distance from the maximum effective range of the CSW unit weapon to the threat location;

SINANG, COSANG = the sine and cosine, respectively, of the angle in the X-Y plane between the CSW unit's current location and (XT, YT); and

XT, YT = coordinates of the location of the enemy threat position for the CSW unit.

RESTRICTIONS: Subroutine CSWDES is called only from subroutine CSWGRD.

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

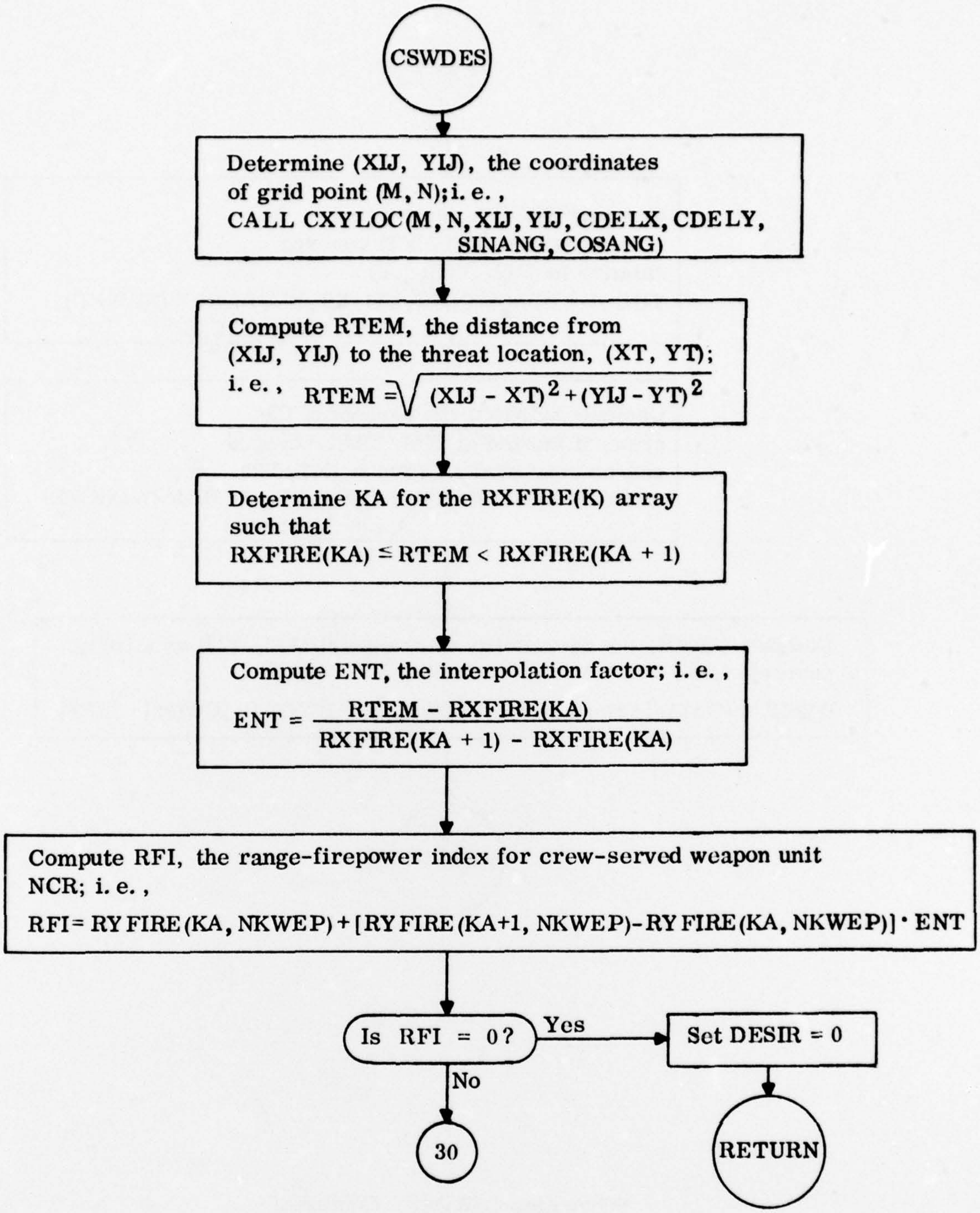
DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED: CXYLOC LOSCMP

COMMON AREAS REFERENCED:

CSWDF1	CW	RXFIRE
CSWIND	HCREW	RYFIRE

STORAGE: $4BE_{16} = 1214_{10}$ bytes



Subroutine CSWDES: Crew-Served Weapons--Fire Positon Desirability (Dismounted Mode)

30

Compute TGCOV, the percent of the threat position, (XT, YT) covered relative to (XIJ, YIJ); i. e.,

TGCOV= LOSCMP(XIJ, YIJ, XT, YT, HCREW(NKWEP), 3., 0., 0, . FALSE.)

Compute CSWCO, the percent of the crew, if located at (XIJ, YIJ), covered and/or concealed relative to (XT, YT); i. e.,

CSWCO= LOSCMP(XT, YT, XIJ, YIJ, 3., HCREW(NKWEP), 0., 0, . FALSE.)

Compute DESIR, the desirability of grid point (XIJ, YIJ) as a firing position; i. e.,

DESIR = [CW(1)] · [1 - TGCOV] + [CW(2)] · [CSWCO] + [CW(3)] · [RFI]

RETURN

Subroutine CSWDES: Continued

Subroutine CSWGRD

PURPOSE: Subroutine CSWGRD determines the primary and alternative firing position, and the optimal route to the primary firing position, for a dismounted crew-served weapon unit.

CALLING SEQUENCE:

CALL CSWGRD(XT, YT, WIDTH)

where

(XT, YT) = coordinates of the location of the enemy threat position for the CSW unit,

WIDTH = width of fire position selection grid.

RESTRICTIONS: Subroutine CSWGRD is called only from subroutine CSWCON.

METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DEFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

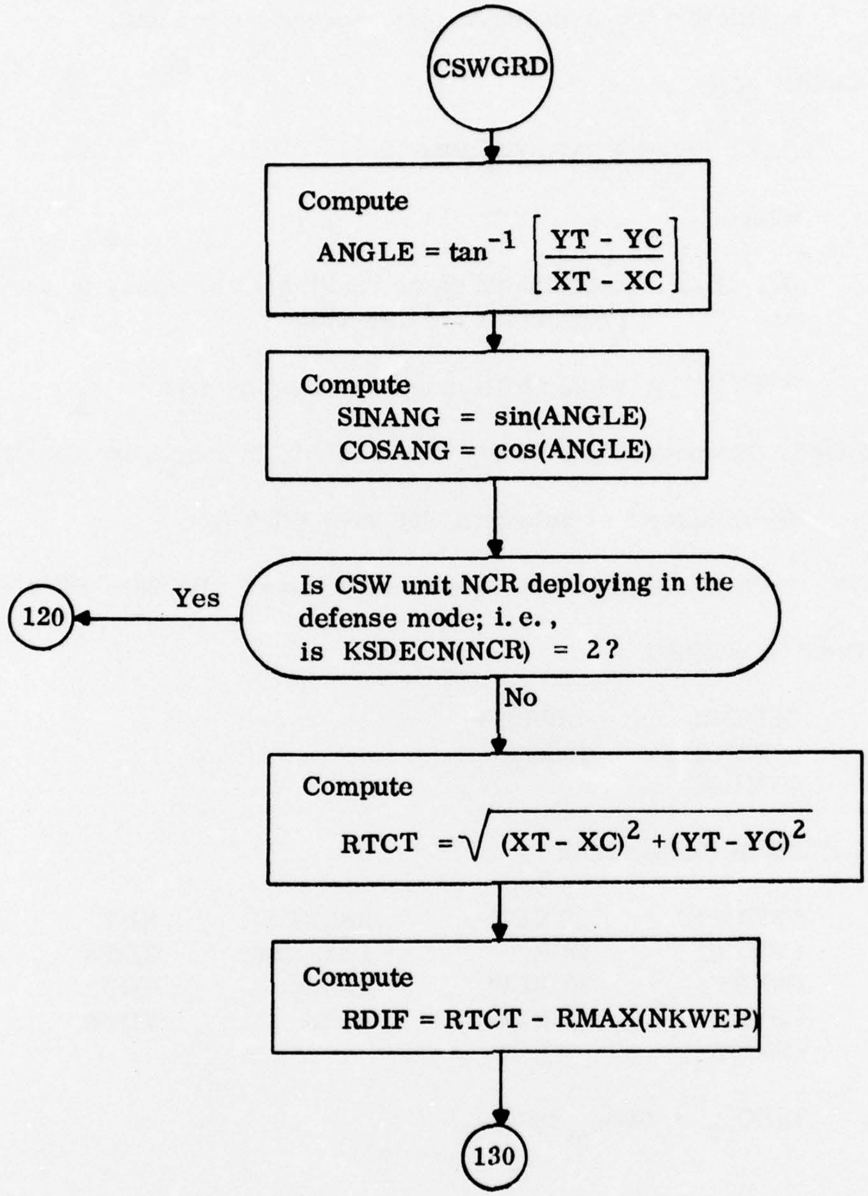
SUBROUTINES REQUIRED:

CSRTSL	LOSCMP
CSWDES	MXDSR
CXYLOC	

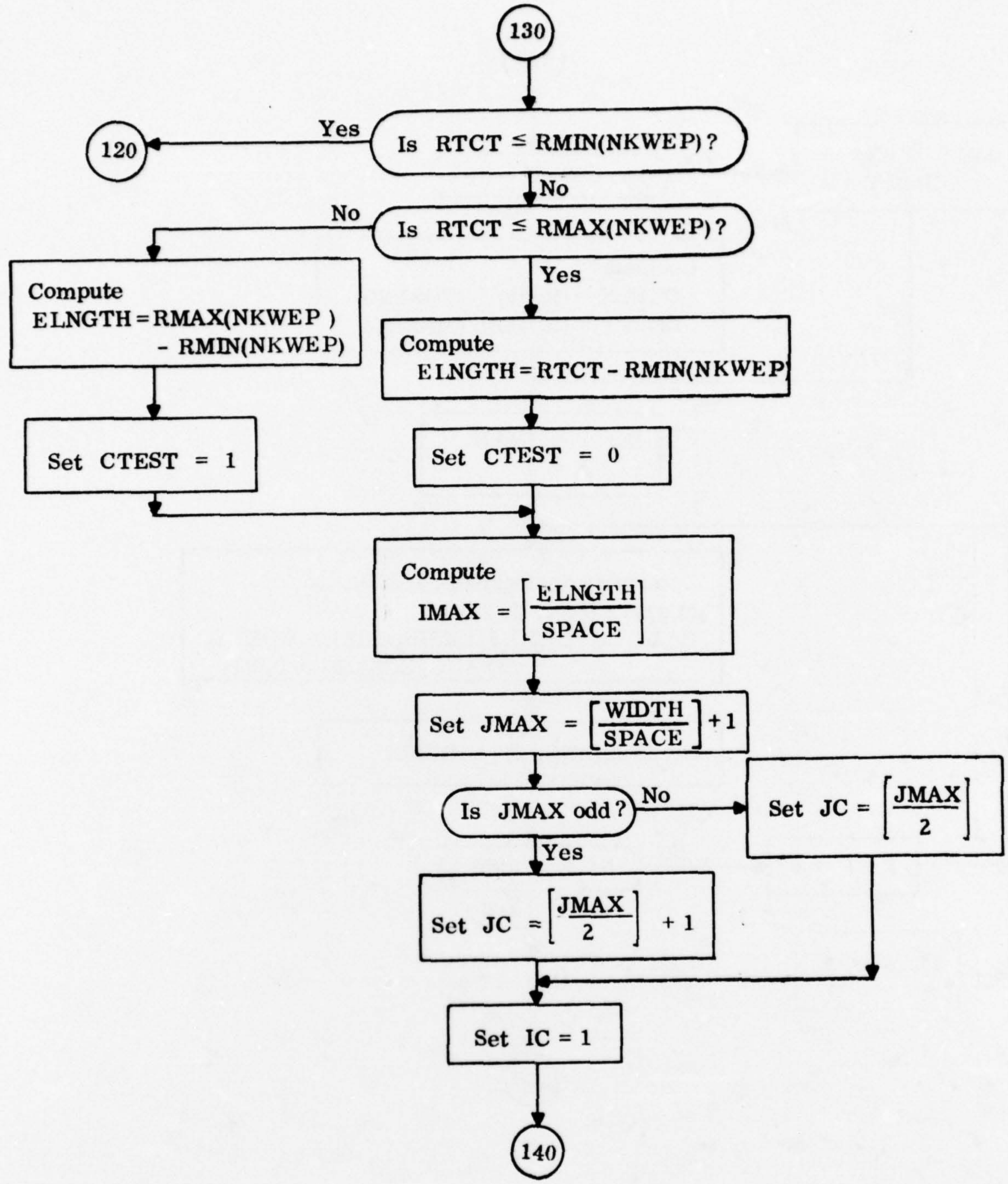
COMMON AREAS REFERENCED:

CSWDF1	CYCUR	KSDECN	XDF
CSWIND	DCW	LAPCCW	XDFA
CSXRT	HCREW	RMAX	YDF
CSYRT	KLCPE	RMIN	YDFA
CXCUR			

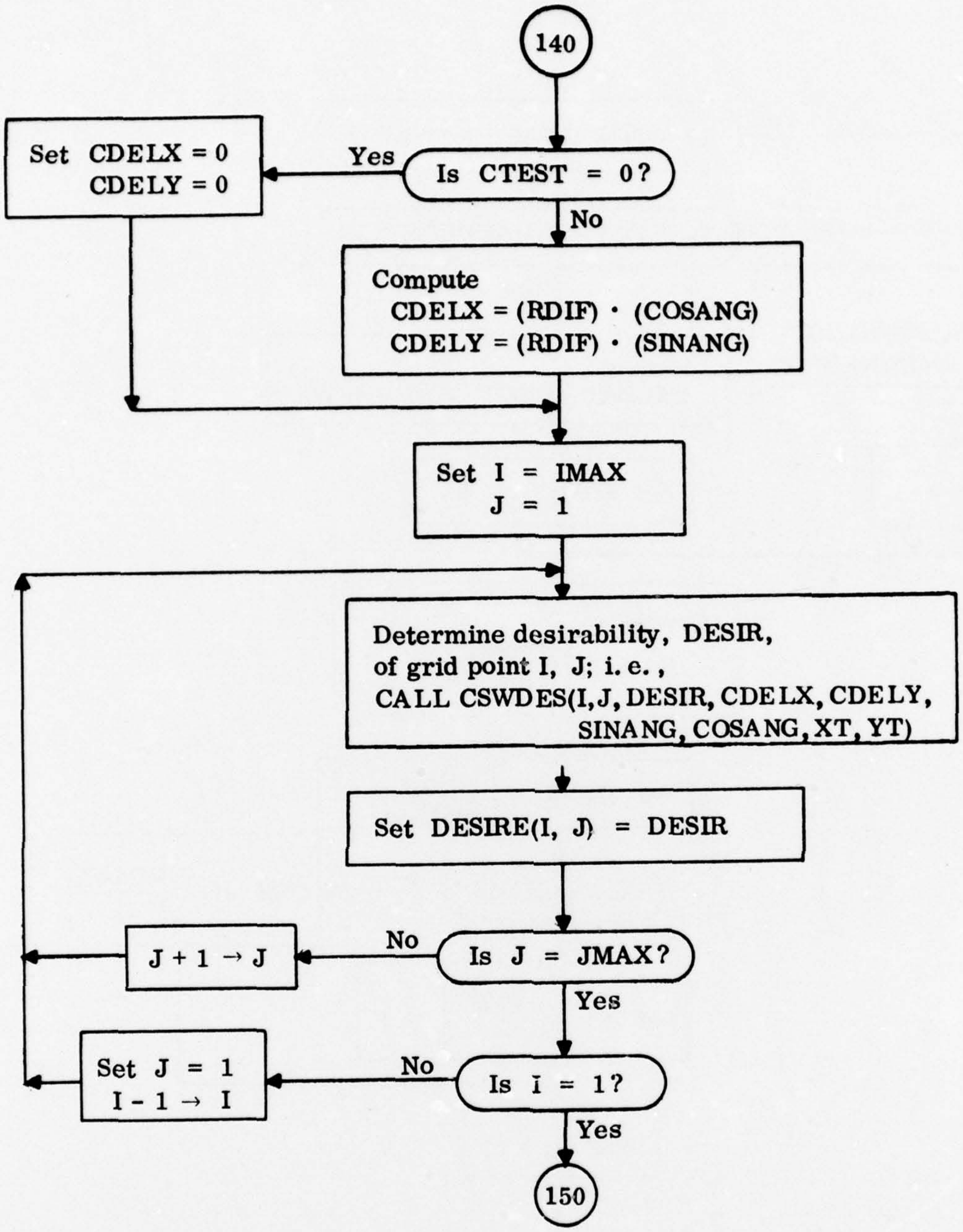
STORAGE: $19EO_{16} = 6368_{10}$ bytes



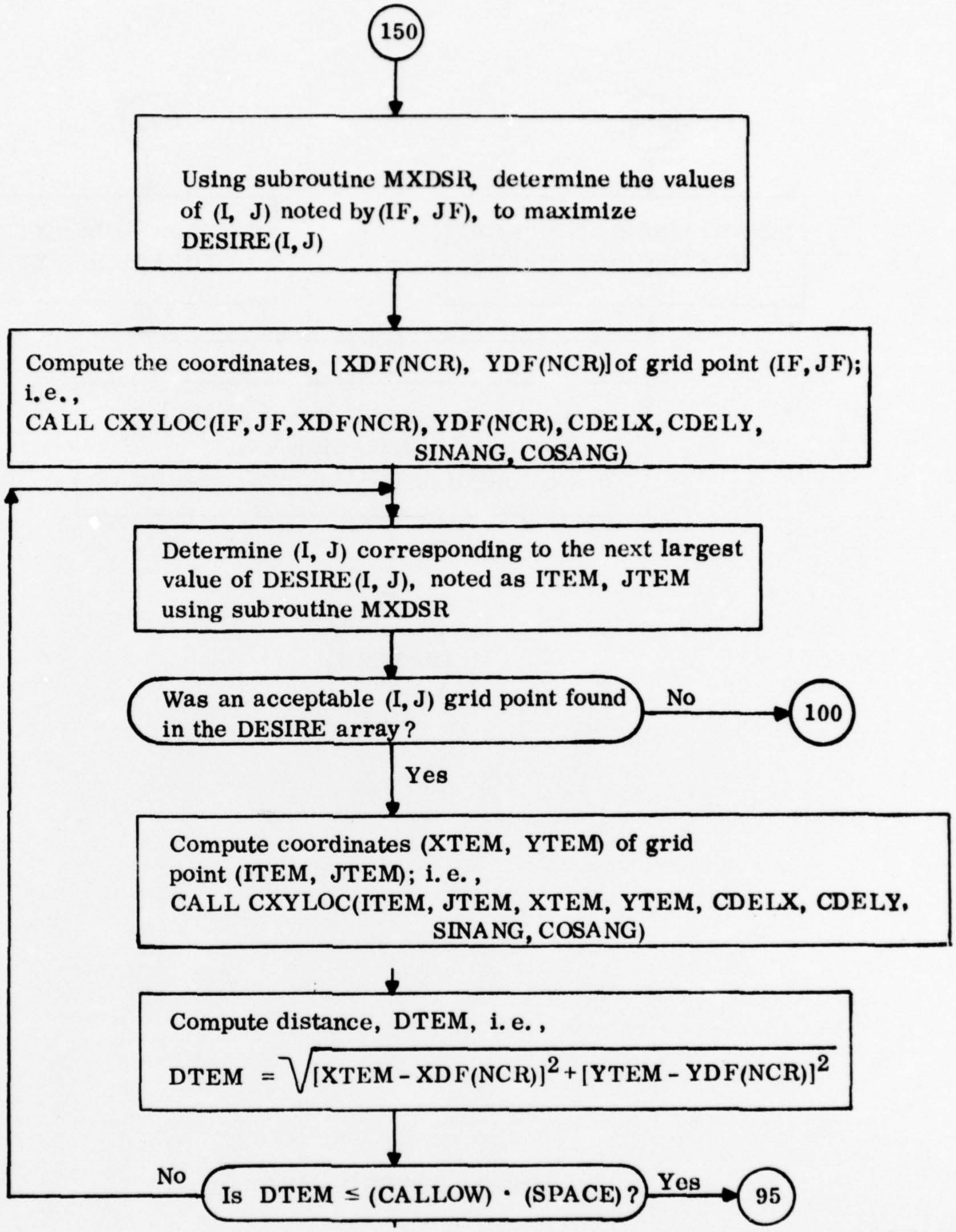
Subroutine CSWGRD: Fire Position Determination
(Dismounted Crew-Served Weapons)



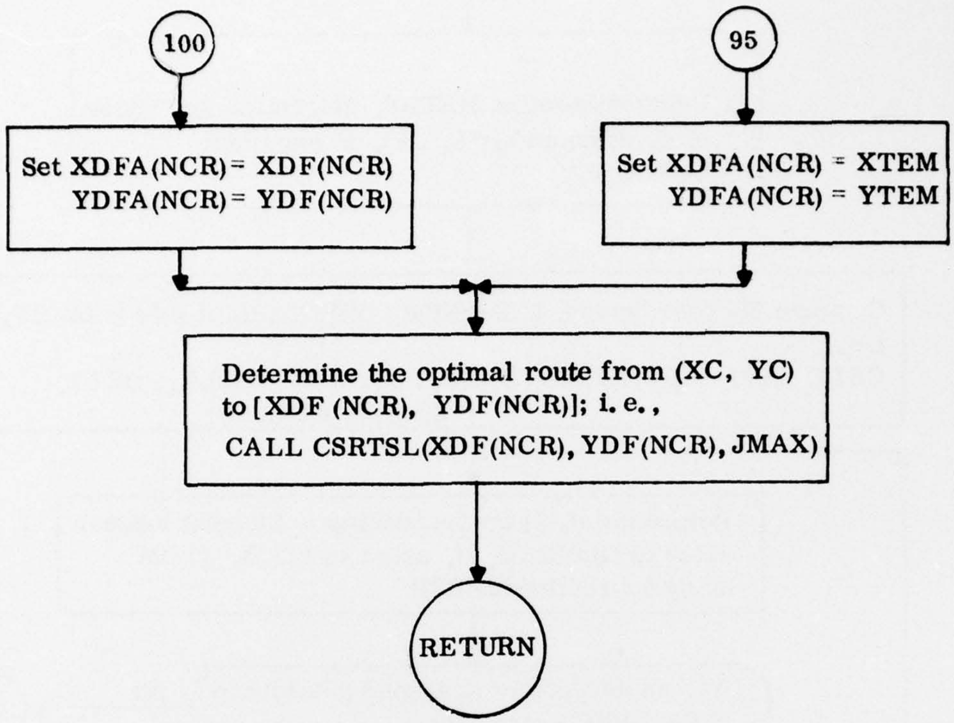
Subroutine CSWGRD: Continued



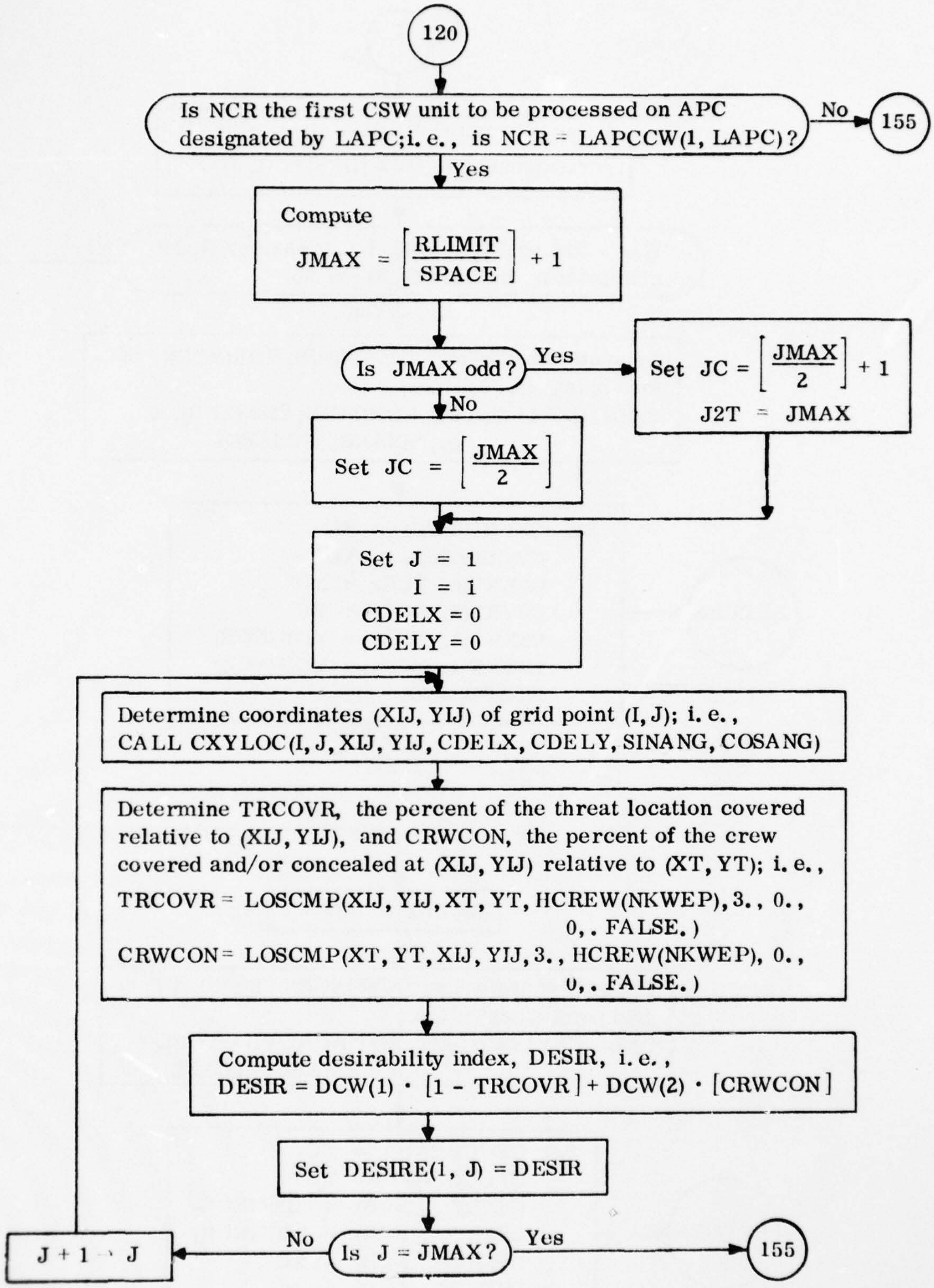
Subroutine CSWGRD: Continued



Subroutine CSWGRD: Continued



Subroutine CSWGRD: Continued



Subroutine CSWGRD: Continued

155

Using subroutine MXDSR, determine JF corresponding to $\underset{J}{\mathrm{MAX}}\,[\mathrm{DESIRE}(1, J)]$

Was a fire position found; i. e., was any (1, JF) grid point in the DESIRE array ≥0. ?

No

Yes

Compute coordinates, [XDF(NCR), YDF(NCR)], of grid point, (1, JF); i. e.,
CALL CXYLOC(1, JF, XDF(NCR), YDF(NCR), 0.,
0., SINANG, COSANG)

Set CXCUR(NCR) = XC
CYCUR(NCR) = YC
CSXRT(2, NCR) = XC
CSYRT(2, NCR) = YC
CSXRT(1, NCR) = XDF(NCR)
CSYRT(1, NCR) = YDF(NCR)
KLCPE(NCR) = 1

RETURN

Set J2T = J2T + 1

Compute coordinates, [XDF(NCR), YDF(NCR)], of grid point (1, J2T); i. e.,
CALL CXYLOC(1, J2T, XDF(NCR), YDF(NCR), 0.,
0., SINANG, COSANG)

Set CXCUR(NCR) = XC
CYCUR(NCR) = YC
CSXRT(1, NCR) = XDF(NCR)
CSYRT(1, NCR) = YDF(NCR)
CSXRT(2, NCR) = XC
CSYRT(2, NCR) = YC
KLCPE(NCR) = 1

RETURN

Subroutine CSWGRD: Continued

Subroutine CXYLOC

PURPOSE: Subroutine CXYLOC determines the X-Y battlefield coordinates of a fire position selection grid point for a dismounted crew-served weapon unit.

CALLING SEQUENCE:

CALL CSYLOC(I, J, XIJ, YIJ, CDELX, CDELY, SINANG, COSANG)

where

(I, J) = designator for the fire position selection grid point for which the X-Y battlefield coordinates are to be determined;

(XIJ, YIJ) = the X-Y battlefield coordinates of grid point (I, J);

(CDELX, CDELY) = the X and Y distance, respectively, of the distance from the maximum effective range of the CSW unit weapon to the threat location;

SINANG = sine of the angle in the X-Y plane between the CSW unit's current location (XC, YC) and the threat location;

COSANG = cosine of the angle in the X-Y plane between the CSW unit's current location (XC, YC) and the threat location.

RESTRICTIONS: Subroutine CXYLOC is called only from subroutines CSWDES and CSWGRD.

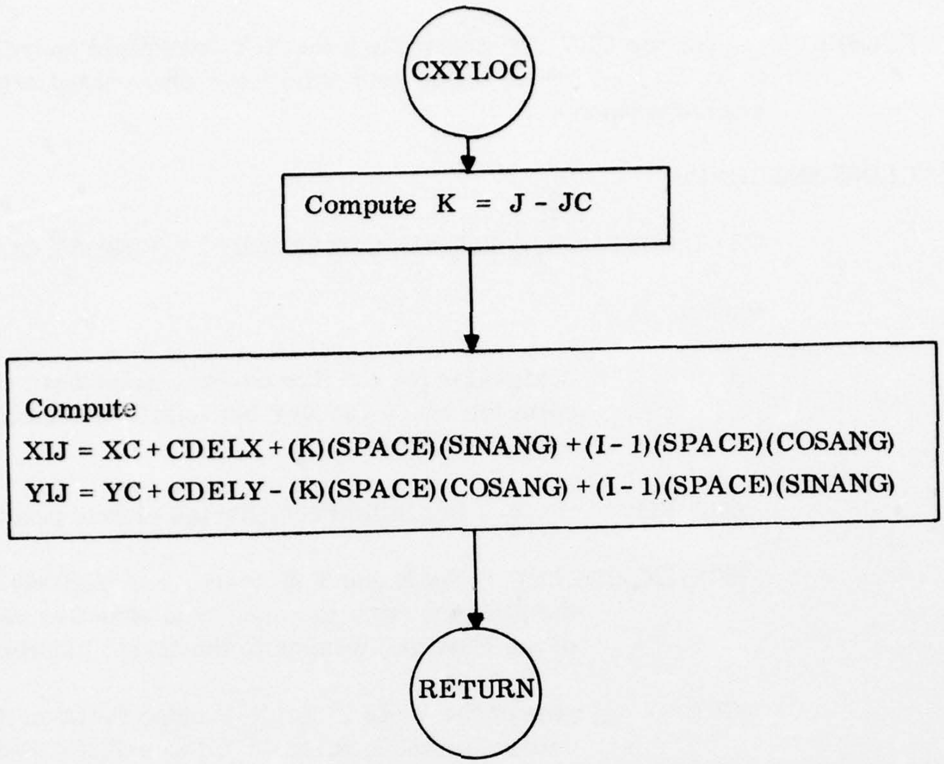
METHOD: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

DFINITION OF VARIABLES: See Chapter 5 of Volume 2, RF 2376 FR-2 (U).

SUBROUTINES REQUIRED: None

COMMON AREAS REFERENCED: CSWDF1 CSWIND

STORAGE: $29A_{16} = 666_{10}$ bytes.



Subroutine CXYLOC: Crew-Served Weapons--Determination of Fire Position Grid Coordinates (Dismounted Mode)